

Received May 9, 2020, accepted May 17, 2020, date of publication May 22, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2996595

# Static and Dynamic Community Detection Methods That Optimize a Specific Objective Function: A Survey and Experimental Evaluation

**KAMAL TAHA** , (Senior Member, IEEE)

Department of Electrical and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates


e-mail: kamal.taha@ku.ac.ae

**ABSTRACT** Most current survey papers classify community detection methods into broad categories and do not draw clear boundaries between the specific techniques employed by these methods. We survey in this paper all fine-grained community detection categories, the clustering methods that fall under these categories, and the techniques employed by these methods for optimizing each objective function. We provide methodology-based taxonomies that classify static and dynamic community detection methods into hierarchically nested, fine-grained, and specific classes. We classify the methods into the objective function they optimize. Each objective function class is classified into clustering categories. Each category is further classified into clustering methods. Methods are further classified into sub-methods and so on. Thus, the lowest subclass in a hierarchy is a fine-grained and specific method. For each method, we survey the different techniques in literature employed by the method. We empirically and experimentally compare and rank the different methods that fall under each clustering category. We also empirically and experimentally compare and rank the different categories that optimize a same objective function. In summary, the block-based, top-down divisive-based, random walk-based, and matrix eigenvector-based methods achieved good results. Finally, we provide fitness metrics for each objective function.

**INDEX TERMS** Clustering, community detection, objective function.

## I. INTRODUCTION

Community detection is an essential objective in graph mining. A community can be defined as a group of similar and densely connected vertices that are sparsely connected with the remaining vertices in the network. Each community has a certain structure that reflects the degree of interaction between its members. Such structure is analyzed for gaining insight into the degree of dynamicity between the members. In a social structure setting, the social network is clustered to reflect social unities such as families, colleagues, villages, and social groups. In biomedical setting, densely connected vertices in metabolic networks are examined for determining functionally related units [122], [124], [125]. In information forensic setting, densely connected vertices in criminal networks are examined for determining criminal organizations [126]. In information systems setting, densely connected vertices are examined for purposes such as business potentials [127].

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman .

Communities can have different domains, properties, structures, hierarchical organizations, and layers, which led to numerous perspectives and community detection methods to be proposed. This, in turn, resulted in unclear boundaries and overlaps between these methods, which has necessitated community detection surveys to provide not only comprehensive, but also fine-grained and specific categorizations of the methods. Unfortunately, most current survey papers classify clustering methods into broad categories and do not draw clear boundaries between the specific techniques employed by these methods [56]. Most of them categorize algorithms into broad two classes [18], [44], [54], three classes [20], [142], four classes [91], or five classes [92], [93], [94], [137]. Most of these papers perform the categorizations in the independence of the following: (1) the objective functions, which the methods seek to optimize, and (2) the broad clustering categories, under which the methods fall.

Many survey papers categorized community detection methods based on the types of their algorithms. Harenberg *et al.* [54] categorized algorithms into two classes:

detect disjoint communities and detect overlapping communities. The authors provided an empirical review of the algorithms in the two classes. Papadopoulos *et al.* [94] categorized algorithms into five methodological classes: divisive-based, vertex-based, model-based, optimization-based, and cohesive subgraph-based. Porter *et al.* [92] categorized algorithms into five classes: modularity optimization-based, centrality-based, local search-based, spectral partitioning-based, and physics-based. Yang *et al.* [142] categorized algorithms into three classes: heuristic-based, optimization-based, and similarity-based. Santo [113] categorized algorithms into eight classes: statistical inference-based, divisive-based, dynamic-based, modularity-based, multi resolution-based, spectral-based, overlapping-based, and  $k$ -means-based. Pons [93] categorized algorithms into five classes: random walk-based, agglomerative-based, separative-based, classical-based, and miscellaneous approaches.

*Significance:* Most current survey papers classify community detection methods into broad categories and do not draw clear boundaries between the specific techniques employed by these methods [56]. This may lead to the following problems: (1) the misclassification of unrelated methods/techniques into a same clustering category, and (2) the exhibition of metrics that measure the quality of methods optimizing a same objective function to different qualitative behaviors (these behavior variations can vanish, if the metrics are applied to methods that fall under a same *fine-grained* class/category). To overcome these limitations, we introduce a methodology-based taxonomy that classifies static and dynamic community detection methods into hierarchically nested, fine-grained, and specific classes. This is the first paper, to the best of our knowledge, that classifies community detection methods based on the following: (1) the objective functions they attempt to optimize, and (2) the clustering categories, whose underlying techniques are employed by these methods.

Other survey papers categorized clustering methods for specific type of networks and communities. Giannini [44] was the first to categorize community detection algorithms for Semantic Web data. Xie *et al.* [137] compared the accuracy of fourteen state-of-the-art methods for detecting overlapping communities. Orman *et al.* [91] compared eight disjoint community detection algorithms using the topological properties of the communities detected. Crampes and Plantié [20] categorized algorithms according to the types of output and input data. Coscia *et al.* [18] classified community detection algorithms according to the definition of the adopted community. Malliaros and Vazirgiannis [79] categorized community detection algorithms using a methodology-based taxonomy. To overcome the above limitations, we introduce in this paper a

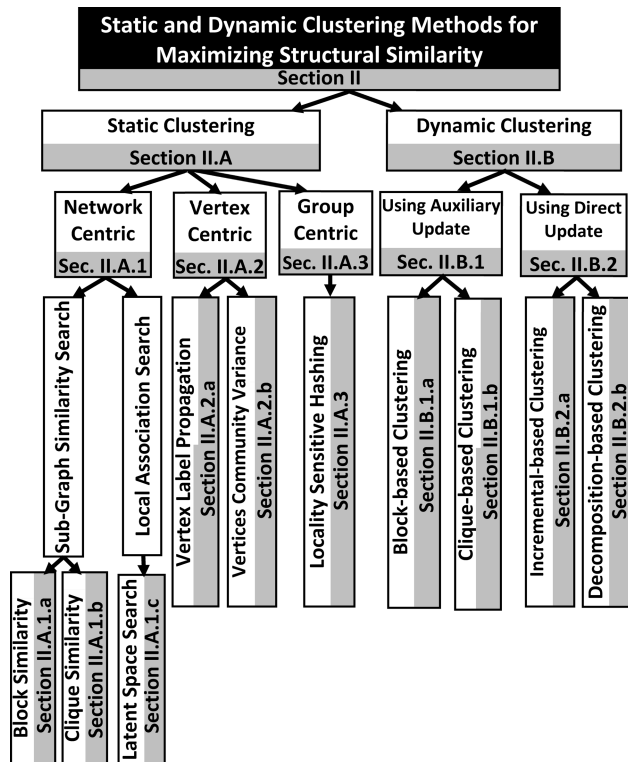
comprehensive survey on static and dynamic community detection categories, the clustering methods that fall under these categories, and the techniques that fall under the methods. We classify methods into hierarchically nested, fine-grained, and specific classes, as follows:

- 1) Classifying classes into the following four objective functions: maximizing internal density, maximizing structural similarity, maximizing dynamic similarity, and maximizing partitions separability.
- 2) Classifying each objective function into the following clustering categories: vertex centric, network centric, group centric, and hierarchy centric (for static clustering), and auxiliary update and direct update (for dynamic clustering).
- 3) Classifying each clustering category into methods.
- 4) Classifying a method into sub-methods, and so on.

Thus, the lowest subclass in a hierarchy is a fine-grained and specific clustering method: the classifications resulted in 31 fine-grained methods. For each method, we surveyed the different techniques in literature employed by the method. We empirically and experimentally compared and ranked the methods that fall under each clustering category. We also empirically and experimentally compared and ranked the different categories that optimize a same objective function. The contributions of this paper are summarized as follows:

- 1) Providing the following methodology-based taxonomy, which hierarchically classifies static and dynamic clustering methods into fine-grained classes: objective functions  $\rightarrow$  clustering categories  $\rightarrow$  clustering methods  $\rightarrow$  clustering sub-methods  $\rightarrow \dots \rightarrow$  clustering sub-methods.
- 2) Surveying the different static and dynamic clustering methods that fall under each clustering category and objective function in the provided taxonomy.
- 3) Discussing the techniques employed by 31 fine-grained methods for detecting clusters.
- 4) Empirically and experimentally comparing and ranking the different methods that fall under each category.
- 5) Empirically and experimentally comparing and ranking the categories that optimize an objective function.
- 6) Providing the fitness metrics for objective functions.

The paper is organized as follows. In Sections II-V, we review and describe the static and dynamic clustering methods that maximize the structural similarity, static and dynamic clustering methods that maximize the internal density, static clustering methods that maximize the partition separability, and static methods that maximize the dynamic similarity, respectively. In Section VI, we provide the fitness metrics for clustering objective functions. In Sections VII and VIII, we evaluate the static community detection methods empirically and experimentally, respectively. In Section IX, we evaluate the dynamic community detection methods empirically and experimentally. We provide our conclusions in Section X.



**FIGURE 1.** A methodology-based taxonomy, which hierarchically classifies the static and dynamic community detection methods that optimize the structural similarity objective function into clustering categories, clustering methods, and clustering sub-methods.

## II. STATIC AND DYNAMIC CLUSTERING METHODS THAT MAXIMIZE THE STRUCTURAL SIMILARITY

We review and discuss in this section the static and dynamic clustering categories and methods that optimize the structural similarity objective function. Fig. 1 presents our methodology-based taxonomy, which hierarchically classifies the static and dynamic community detection methods that optimize the structural similarity objective function.

### A. STATIC CLUSTERING CATEGORIES AND METHODS

#### 1) NETWORK CENTRIC CLUSTERING CATEGORY

##### a: METHODS THAT APPLY BLOCK-BASED SUB-GRAPH SIMILARITY HEURISTIC SEARCH

These methods partition vertices into subgroups called blocks. They assume that all vertices of a block are stochastically equivalent and have the same probabilities of interaction with each other [3], [85]. They provide a generalization model of the blockmodel [36] by allowing for data variability for detecting community structures. These methods aim at overcoming the limitation of the blockmodel of ignoring the variation in vertex degree. Towards this, they estimate the degrees of interaction between each vertex and other vertices. Peixoto [99] proposed a stochastic block model that detects community structure by shifting groups of nodes simultaneously instead of individual nodes. The model employs an adjusted version of the Markov Chain Monte Carlo (MCMC)

scheme to simultaneously move multiple nodes at each step. The movements are performed by rearranging, splitting, and merging groups of nodes. The proposed scheme samples partitions from a posterior distribution to improve the mixing time of the MCMC in empirical situations.

Karrer and Newman [66] proposed a heuristic algorithm that is a degree-corrected version of the blockmodel. Results showed that the proposed algorithm demonstrated an improved community detection in complex networks. Xu and Hero [138] proposed an extension of the stochastic block model. It is a statistical model for dynamic and time evolving networks. It can be used for posteriori blockmodeling or a priori. Chen and Saad [16] proposed a method inspired by matrix blocking, which is the process of reordering the columns and rows of a matrix in such a way that the blocks alongside the diagonal represent dense subgraphs.

##### b: METHODS THAT APPLY CLIQUE-BASED SUB-GRAPH SIMILARITY HEURISTIC SEARCH

These methods build communities from cliques. A clique is a fully connected subset of  $k$  vertices. If two cliques share  $k - 1$  vertices, they are adjacent to each other. The methods construct a community from a union of cliques that are reachable from one another. A clique can reach another one through a series of adjacent cliques. The Clique Percolation Methods (CPM) fall under this category. Qian *et al.* [103] proposed a model that detects overlapping community structure by maximizing the similarities between clique connections. The model initializes the structure of a network and quantifies the connection similarity of a community using the concept of maximum clique. The concept is based on the sharing of the connections and nodes between different communities. Based on these sharing, closely connected communities are merged to identify the overlapping communities.

Qian *et al.* [104] proposed a method for detecting community structure from a heterogeneous network by employing the maximum bipartite clique technique. The network is clustered into maximal groups. The most influential largest two groups are used as initial communities. These initial communities are then expanded based on their similarities with neighboring nodes. The neighboring nodes of each initial community are compared with other nodes to determine whether they are related. Nodes are divided accordingly. Yuan *et al.* [143] proposed a method for detecting community structure by treating a  $k$ -clique percolation community as a union of *maximal* cliques. Given a set of query nodes, the method identifies a  $k$ -clique percolation community that: (1) has the maximum  $k$  value, and (2) contains the query nodes. A clique percolation community that satisfies the two conditions is considered the densest.

Palla *et al.* [97] proposed a tool based on CPM for detecting overlapping communities and identifying the general characteristics of networks in society and nature. The authors assumed that a group of  $k$  cliques that shares at least  $k - 1$  vertices with one another constitutes a community. Edges represent the intensities of the overlap of cliques in a network.

To detect overlapping communities, the tool first identifies all  $k$  cliques in a network. Then, it identifies a clique-clique overlap matrix. Farkas *et al.* [37] proposed a method for weighting  $k$ -clique communities called Weighted Clique Percolation Method. The authors defined a term called  $k$ -clique severity as the average of  $k*(k-1)/2$  link weights. They defined also a term called Directed Clique Percolation Method to refer to the directed  $k$ -clique communities that have direct link between each two vertices in a  $k$ -clique. Kumpula *et al.* [65] proposed a clique-based method, where each subset of  $k$  vertices is processed individually. First, the method identifies all  $k-2$  cliques that share the adjacency of two endpoints. Second, it identifies the connected parts in the  $k-1$  cliques. Each vertex represents a  $k$ -clique and the other represents a  $k-1$  clique.

### c: METHODS APPLY LATENT SPACE LOCAL ASSOCIATION SEARCH

These methods assume that densely connected vertices are likely to occupy latent positions close to each other [52], [114]. That is, the interactions among vertices depend on their positions in the latent space. Positions are identified using a maximum likelihood estimation. Sankararaman and Baccelli [116] proposed a community detection method in spatial random graphs, which is a planted-partition version of the random connection model. Each node is associated with two labels, which are valued community label and valued location label. Depending on the Euclidean distance between nodes and their community labels, edges and labels are selected randomly and independently. Thus, the accuracy of a detected community structure relies on the random graph's observation and the location of the spatial labels on nodes.

Sarkar and Moore [114] proposed a framework that can turn a static relationship model into a dynamic one that accounts for moving friends in and out a community. The framework can associate each vertex with a point in  $p$ -dimensional Euclidean latent space. Handcock *et al.* [52] argued that the Euclidean distance between two individuals in the latent social space reflects the probability of a tie between them. The authors considered the approximate conditional Bayes factors for identifying the number of communities in a network. Reichardt and Bornholdt [106] proposed a Potts model consisting of spins placed on a lattice. The lattice is a two-dimensional rectangular Euclidean. It can be generalized to other dimensions or lattices. Each vertex is in one of the spin states.

## 2) VERTEX CENTRIC CLUSTERING CATEGORY

### a: METHODS THAT APPLY VERTEX LABEL PROPAGATION BASED ON STOCHASTIC PROCESS

These methods perform clustering after propagating the community labels of a selected subset of vertices to other vertices. Most of these methods select some vertices and assigns their community label to their neighbors. Jiang *et al.* [60] proposed

a method for detecting community structure based on Label Propagation Algorithm (LPA). The method clusters together highly influential users that have similar interests. This leads to minimizing the negative impact of the inclusion of users who may have less similar interests. The method is composed of two modules, one for scoring users' interests and the other for clustering users based on these scores. In each iteration of the label updates, selecting the majority of the adjacent nodes are held by the updated label. This is done by selecting the majority of its adjacent nodes.

Bhatt *et al.* [13] proposed a label propagation-based method for detecting the structure of a community based on the context describing it. The method predicts the common context that summarize a potential community's nodes. First, each node is labeled with contextual information that describes multiple domain-specific concepts. The proposed algorithm optimizes the following two tasks iteratively: (1) optimizing the assignment of a community's label without changing the community's context, (2) optimizing the assignment of a community's context without changing the community's labels constant. The first task is achieved by proposing a contextual similarity measure for measuring the similarities between nodes. The second task is achieved by balancing informativeness and purity.

Mehrabi *et al.* [81] proposed a label propagation-based method for detecting the structure of a community. The method attempts to mitigate the problem of sparsely connected nodes in a network by assigning loosely connected nodes to their appropriate communities. This leads to assigning insignificantly labeled users to their appropriate significant communities. The method employs an unsupervised learning mechanism for detecting communities using modularity and network attributes.

Li *et al.* [74] proposed a label propagation-based method that applies motif mining to identify the higher order structure of a network for detecting the structure of the network's communities. The method detects triangle motifs in a network to identify the structural characteristics in the network. The following are the sequential processing steps taken by the method: (1) identifying the motif of interest, (2) constructing a hypergraph to encode the higher order connections, (3) designing a re-weighted network, and (4) applying a voting strategy to update the labels of nodes. Chin and Ratnavelu [27] proposed a label propagation-based method that updates unassigned nodes synchronously and assigned nodes asynchronously. The method employs a similarity score measure during the propagation process to identify the initial communities and to break ties. A community that reaches a specific strength threshold is exempted from the merging and procedure. This process is repeated iteratively until labels' convergence is achieved.

The Speaker-listener Label Propagation Algorithm (SLPA) proposed by Raghavan *et al.* [108] is an extension of the Label Propagation Algorithm (LPA) proposed by Xie and Szymanski [135]. In this algorithm, each vertex is initially considered as a separate community. Then, another vertex is

selected as a listener. A label is propagated to each neighbor (*speaker*) of the listener. Propagated labels are selected randomly with probabilities proportional to their frequencies in the memories of the speakers sent them. A listener selects the most common labels it received, whose probability distributions are greater than a given threshold. These common labels form a community. Tasgin *et al.* [123] proposed a method that selects some vertices and assigns their community label to their neighbors. It detects communities from the labels of a small number of seed vertices. Gong *et al.* [43] advocated using similar methodology and recommended applying it to 20% of the vertices. Blum and Mitchell [11] introduced a model that augments training dataset by labels that are incompletely related to the dataset. Sindhwani and Niyogi [112] proposed an extension of regularization algorithms that require unlabeled training datasets to be available in multiple views.

#### *b: METHODS THAT CONSIDER VERTICES COMMUNITY VARIANCE*

These methods are based on the assumption that if the community variance of some vertex  $u$  and a set  $S$  of vertices is small,  $u$  and  $S$  should belong to the same community. Žalik and Žalik [145] proposed a method for detecting community structure using node attraction in local processing and learning. The degree of a community's variance is determined based on the degrees of association between each neighboring nodes in the community. The degree of belonging of a node to a community is determined by the degrees of some of the community's nodes that are attracted to this node. This, in turn, increases the modularity of communities.

Let  $N$  be the set of neighboring vertices to a vertex  $u$ . The community variance proposed by Tasgin *et al.* [123] is the ratio of: (1) the number of communities that include both  $u$  and a vertex that belongs to  $N$ , and (2) the number of neighbors of  $u$ . Shang *et al.* [111] employed simulated annealing method as a local search for community variance, as follows. Let  $Q(C_1)$  be the modularity of a cluster  $C_1$  that contains a subset  $S_1$  of vertices with the highest modularity. Let  $Q(C_2)$  be the modularity of a cluster  $C_2$  that contains the subset of vertices with the highest modularity less than  $S_1$ . Let  $r \in [0, 1]$  be a randomly generated parameter. If the community variance.  $CV = Q(C_1) - Q(C_2) > r$ ,  $S_1$  will be assigned to  $C_1$ . The method proposed by Gong *et al.* [43] performs local searches iteratively. At each iteration, if a vertex  $v$  that belongs to a cluster  $C_x$  achieves the best fitness value with a cluster  $C_x \neq C_x$ ,  $v$  is deleted from  $C_x$  and assigned to  $C_x$ .

#### 3) GROUP CENTRIC CLUSTERING CATEGORY

Most of the methods that fall under this category employ locality sensitive hashing clustering techniques. They employ the hashing-based techniques to approximate the nearest neighbors to given vertices. They cluster together the sets of vertices whose neighborhoods are overlapped. Macropol and Singh [78] proposed a probabilistic clustering method called

TopGC that identifies connected clusters in a network using a hashing-based technique called MinHash. The technique estimates the similarity between two sets. Two sets are considered similar, if their neighborhoods are overlapping. These sets are clustered together. The strength of a cluster resulted from the merging of sets is assessed by measuring the ratio of: (1) the sum of the weights of edges in the cluster, and (2) the number of edges multiplied by the original cluster size.

### **B. DYNAMIC CLUSTERING CATEGORIES AND METHODS**

#### 1) USING AUXILIARY UPDATE CLUSTERING CATEGORY *a: BLOCK-BASED CLUSTERING METHODS*

Xubo *et al.* [139] proposed a block-based method for detecting community structure from temporal networks, whose data can change or evolve over time. The method employs a reduction strategy using sampling. Then, it rearranges the original temporal network. First, an auxiliary representation of the original network is extracted by sampling nodes. Then, each detected pattern is modeled into a different community. Each durable temporal state is regarded as a community.

Lin *et al.* [76] proposed a stochastic block model called FaceNet for detecting community structure and a probabilistic model for capturing the evolutions of communities in dynamic networks. In this model, the structure of a community at a specific timestamp  $t$  is identified by the combination of: (1) the prior distribution of historic community structures, and (2) the observed data at  $t$ . The probabilistic model assigns soft community memberships to the nodes. Angel *et al.* [5] proposed a method that maintains a dense block subgraphs caused by quantifying the maximum change resulted from updating edges weights. The method can compute dense subgraphs incrementally by keeping a small number of sparse subgraphs. It employs a dense subgraph index that decreases the consumption of memory.

#### *b: CLIQUE-BASED CLUSTERING METHODS*

Duan *et al.* [31] proposed a clique-based method that regards social networks' dynamics as a change stream. The method employs an incremental  $k$ -clique clustering algorithm. The algorithm adopts an auxiliary updating procedure based on local depth first search forest. Cazabet *et al.* [25] proposed a dynamic clique-based method for detecting community structure. The method adds new edges to an auxiliary network at a given time step. The minimal community is considered to be a predefined clique pattern with 3, 4, or more nodes. Every time a new edge is included in the network, the formation of a minimal community is checked. Palla *et al.* [95] proposed a dynamic percolation clique-based method for detecting overlapping communities that evolve over time. This leads to identifying the relationships that characterize the evolution of overlapping communities. The authors concluded that a community can have a better adaptability, if a large number of its members are able to change its composition dynamically.

## 2) USING DIRECT UPDATE CLUSTERING CATEGORY

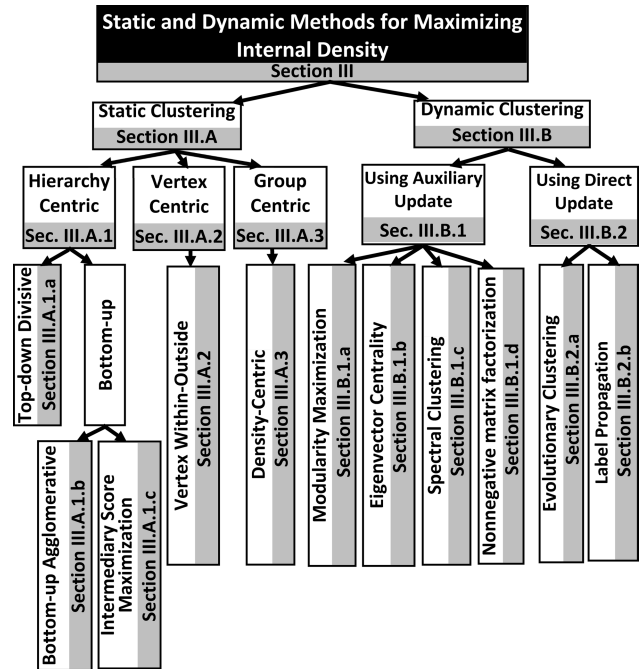
### a: INCREMENTAL-BASED CLUSTERING METHODS

Liu *et al.* [75] proposed an incremental bottom-up method for detecting communities from dynamic graphs. The method adopts a methodology from Link Prediction and Information theory for quantifying a changed node's influence, which helps in identifying the portion of the graph that needs to be recalculated. The proposed algorithm is based on link entropy. Bogdanov *et al.* [15] proposed an incremental sub-interval aggregation methodology for computing and uncovering the highest-scoring temporal subgraph from evolving and weighted edge networks. A temporal subgraph's score is the summation of the weights of edges. The method adopts a filtering methodology for pruning sub-intervals. The search procedure follows an aggregation bottom-up scheme.

Hopcroft *et al.* [55] proposed an agglomerative clustering method for detecting community structure from temporal linked social networks, whose data can change or evolve over time. The structure of the network data is uncovered by averaging the accidental clusters that take place after each run. The structure of a community is estimated by prior distribution and the observed networked data. Falkowski *et al.* [39] proposed a density-based agglomerative incremental method for detecting dense subgroups. The method employs a distance functions for measuring the distance between each two interacting nodes. It also employs a function that updates incremental clustering. The algorithm deals with dynamic datasets using a density-based clustering function. The function applies density-based clustering to graph structures. Gorke *et al.* [47] proposed a cut-based agglomerative incremental method for maintaining the clustering of a changing graph dynamically. The method permits atomic changes in graphs and maintains consecutive temporal smoothness. It keeps updating minimum-cut trees and maintaining the clustering of the graph.

### b: DECOMPOSITION-BASED CLUSTERING METHODS

Rossetti and Rdyn [110] proposed a decomposition-based clustering method for detecting communities from dynamic graphs. At any given date  $d$ , the time steps in the past are considered when detecting communities. The method best suits the online clustering of networks. Aynaud and Guillaume [6] proposed a decomposition-based method for tracking communities between each two successive snapshots of a network's evolution. The method keeps removing random nodes individually and keeps only the largest connected subnetwork. Falkowski *et al.* [40] proposed a method for detecting and analyzing evolving online communities. The method employs statistical and visualization techniques for analyzing evolutions in communities with membership structure. For identifying densely connected subgroups, the method employs a hierarchical-based clustering algorithm based on edge betweenness divisive procedure.



**FIGURE 2.** A methodology-based taxonomy, which hierarchically classifies the static and dynamic community detection methods that optimize the internal density objective function into clustering categories, clustering methods, and clustering sub-methods.

## III. STATIC AND DYNAMIC CLUSTERING METHODS THAT MAXIMIZE THE INTERNAL DENSITY

We review and discuss in this section the static and dynamic clustering categories and methods that optimize the internal density objective function. Fig. 2 presents our methodology-based taxonomy, which hierarchically classifies the static and dynamic community detection methods that optimize the internal density objective function.

### A. STATIC CLUSTERING CATEGORIES AND METHODS

#### 1) HIERARCHY CENTRIC CLUSTERING CATEGORY

The methods that fall under this clustering category build a hierarchical structure of detected partitions based on the topology of the entire network. They assume that if some vertices behave in the same role during interaction, the social status of the individuals represented by these vertices is likely to be similar. They analyze the interaction patterns of vertices to infer their influences and roles.

#### a: TOP-DOWN DIVISIVE METHODS

These methods focus on decomposing a network until a tide partition is attained. Towards this, some of these methods keep removing edges that have high betweenness scores [45]. The betweenness score of an edge is the number of shortest paths that pass through the edge. Ni *et al.* [89] proposed a top-down divisive method for detecting community structure that regards a network's communities as a geometric decomposition. The method employs the underlying principles of the discrete Ricci flow. By determined

heavily traveled edges through Ricci flow process iteratively, the method is able to detect communities. The method proposed by Caldarelli *et al.* [22] keeps removing the inter-community edges with large betweenness scores until several sets of community structures are detected. Then, it aggregates these sets into a final list of detected communities.

The method proposed by Girvan and Newman [45] was also based on removing inter-community edges. The method computes several measures of edge centrality to estimate the importance of edges based on their control of the flow of information in the network. The following are the sequential processing steps taken by the method: (1) computing the centralities of edges, (2) removing the edges with large centralities, (3) re-calculating the centralities of the remaining edges, and (4) repeating steps 2-4. Most of the above methods employ modularity measures to evaluate the quality and strength of their detected communities [19], [86], [87].

#### *b: BOTTOM-UP AGGLOMERATIVE METHODS*

These methods measure the similarities between vertex pairs. They add edges to communities based on the similarities between the vertices at their endpoints. Initially, the search starts from some hierarchy (e.g., a vertex), which is considered as a separate community [17]. These communities are kept being merged (i.e., expanded) until some objective function achieves a local maximum. Zhang *et al.* [146] proposed a bottom-up agglomerative method based on the concept of true-link for detecting community structure. The method transforms the original network's true-link network into link space graph. Then, the method uses signaling processing to identify the link communities. The method merges each two similar sub-communities into point communities during the mapping of link communities.

Bahulkar *et al.* [14] proposed a bottom-up agglomerative method for detecting community structure in criminal networks. The method augments a criminal network containing purposely hidden edges. It can uncover the hidden edges in a network and augments them to the network before detecting communities. It adopts a bottom-up search for detecting communities by optimizing their local modularity. The method proposed by Riedy *et al.* [105] starts by selecting a set of disjoint communities. Then, these communities are kept being merged until a modularity objective is maximized. The method proposed by Clauset *et al.* [23] starts by considering each vertex as a separate community. Then, communities are kept being merged based on their modularity scores until these scores stopped to increase. The method proposed by Clauset *et al.* [23] infers communities based on the topology of the network. It optimizes the modularity function of Newman and Girvan [87] using a greedy technique. The method proposed by Blondel *et al.* [12] starts by employing a local search to select small communities. Then, communities are kept being aggregated until their modularity scores stopped to increase.

#### *c: BOTTOM-UP INTERMEDIARY SCORE MAXIMIZATION METHODS*

These methods keep iteratively adding vertices with high intermediary scores to currently detected communities until some objective function is optimized. Ni *et al.* [88] proposed a bottom-up intermediary score maximization method for detecting local overlapping communities. The method selects a subset of the set of nodes that belong to more than one community. This subset serves as seed nodes. Then, the communities to which these seeds belong are detected. If the fuzzy relation between a given node and a seed node is large enough, the method considers the two nodes belong to a same community. Whang *et al.* [134] proposed a bottom-up intermediary seed expansion method for detecting overlapping communities. The method identifies the best nodes to serve as seeds and greedily expands them to form communities based on a metric. The seeding identification strategy is based on the multi-level weighted kernel  $k$ -means function.

The greedy heuristic method proposed by Jiang and Singh [57] starts by selecting the vertex  $v$  with the maximum weighted degree along with the neighbor  $u$  of  $v$  that has the highest weighted degree among the neighbors of  $v$ . The pair  $u$  and  $v$  is used as a seed of community. Then, the vertices with the highest supports (i.e., intermediary scores) are kept being iteratively added to the current community, until the density of the community reaches a user-defined threshold. Otherwise, the community's edges and the vertices at the endpoints of these edges are removed from the network. The bottom up-based method proposed by Pascal and Latapy [98] assigns a modularity score (i.e., intermediary score) to each traversed vertex for the purpose of cutting the dendrogram in the same manner as fast greedy algorithms. Then, it keeps merging communities based on the outcome of the modularity scores of the vertices after traversing them using random walks.

#### 2) VERTEX CENTRIC CLUSTERING CATEGORY

The methods of this category expect each vertex in a partition to satisfy certain properties, such as adjacency and reachability. Vertices are considered similar, if they share the same connection pattern. This strategy resembles the notion of regular equivalence, where two vertices are considered structurally equivalent, if they share the same neighborhood. The vertex within-outside ties clustering methods falls under this category. In these methods, each vertex in a community is connected to more vertices inside the community than to vertices outside the community [10]. Therefore, removing any link inside the community is unlikely to disconnect it. Gong *et al.* [46] proposed a multi-granularity vertex centric-based method for detecting community structures in social networks. Each network is depicted using a network embedding strategy, which represents each node by its low-dimensional vector representation. If two nodes share the same neighborhood network structures, their embedding is considered similar.

### 3) GROUP CENTRIC CLUSTERING CATEGORY

The methods of this category consider the overall connections inside a partition. They consider a partition acceptable, if it satisfies certain properties (e.g., cohesiveness), even if the connectivity of some of its vertices is low. Cohesiveness characterizes the internal structure of a partition, such as being hard to split into two sub-partitions. The density centric-based clustering methods falls under this category. These methods detect communities based on the density property of partitions. Some of the methods that fall under this category consider a partition  $(V_s, E_s)$  is  $\gamma$ -dense if:  $E_s(V_s(V_s - 1)/2) > \gamma$ , where the partition becomes a clique, when  $\gamma = 1$ .

Fang-Ju [38] proposed a group centric-based clustering method for detecting community structures in social networks. Nodes that are similar are clustered to a same community. The similarity between a pair of nodes is measured based on their common neighbors. If two nodes share a large number of common neighbors, they are assigned to a same community. Modularity parameter is used for measuring the strength of a community. Newman and Park [84] proposed a method called link clustering coefficient, which assumes that short loop links, such as squares and triangles, are likely to be inter-community links, while long loops are across community links. According to the method, short inter-community loop links are likely to increase the density of the community. The method defines the clustering coefficient of a link as the number of squares and triangles that are part of the link. Links with the minimum coefficient are cut off.

## B. DYNAMIC CLUSTERING CATEGORIES AND METHODS

### 1) USING AUXILIARY UPDATE CLUSTERING CATEGORY

#### a: MODULARITY MAXIMIZATION METHODS

Gorke *et al.* [51] proposed a dynamic modularity maximization method for detecting community structure from temporal networks, whose stream changes or evolves over time. The method employs a global greedy procedure as follows. It merges each pair of clusters and computes the increase in modularity of the merged singleton. The procedure is repeated until no further improvement can be made. Pizzuti and Socievole [101] proposed a method for detecting communities in dynamic networks by considering the concept of modularity as a function that needs to be optimized simultaneously on all snapshots. The method uses a cluster-based similarity partition algorithm. Principal clusters are detected by applying a  $k$ -means clustering method. The value of  $k$  is selected in such a way that the modularity of the multi-layer network is maximized.

Shang *et al.* [117] proposed an incremental modularity based method for detecting communities in dynamic networks. First, an initial community is detected statically. Then, incremental updating strategies are performed to detect the dynamic communities. Changes of networks are modeled as sequential increments of edges. Dinh *et al.* [32] proposed a method for detecting maximized modular structure

in dynamic social networks. The method employs previous states' modular structures to adaptively guide the identification of next states' modules. Modules that have negative modularity are merged with their neighboring modules to produce new modules with higher modularity. This process is repeated until the modules that have negative modularity are exhausted.

Gorke *et al.* [48] proposed a method for the heuristic dynamization of current static algorithms to maximize modular structures from dynamic networks. The method quantifies an algorithm's degree of smoothness for transitioning from an output to the next one by comparing its consecutive clustering. Fortunato [42] investigated the applicability of modularity in community detection. The authors found that the intrinsic scale of modularity depends on the number of network's links. They found that the specific structure of a network is irrelevant to the modularity's limit of resolution. Rather, the resolution depends on pairs of communities' degree of interconnectedness.

#### b: EIGENVECTOR CENTRALITY METHODS

Guan and Wu [50] proposed an algorithm that analyzes nodes' context information and historical interaction in social networks. It does so to identify a node's most suitable next-hop node from the pool of its nodes. It assigns a score to each node that reflects its fitness to a community. A node's score is represented by its eigenvector centrality. The algorithm maximizes the dynamic similarity by employing the eigenvector centrality using Bayesian derivation. It adopts the concept of preference similarity between nodes by analyzing their preferences in the transmission process. Nodes' similarities and neighbor information are considered when next hop nodes are measured.

Márton *et al.* [82] proposed a method that controls nodes in a multiplex network to steer it to a desired state. This is because high-centrality nodes may have different influences on the behavior of a network. The method employs rank aggregation techniques to identify the target nodes that maximize the interventions of multi-objective in multiplex networks, based on inter-layer structural correlations. The function of a particular layer and the nodes in the layer are ranked separately based on their centralities. Takaffoli *et al.* [128] proposed a centrality-based framework for modeling and detecting community evolution in dynamic social networks. First, the framework tracks and determines similar communities over time. Then, the evolutions of communities are determined using a series of transitions and events. The framework uses a one-to-one matching procedure to identify the similarity between communities obtained from different snapshots. Asur *et al.* [7] proposed an event-based model for characterizing the evolution of dynamic evolving interacting networks. The model can identify interesting events from interacting non-overlapping snapshots. The authors employed centrality-based behavioral patterns for investigating the impact of influence maximization in cluster evolution.



### c: SPECTRAL CLUSTERING METHODS

Ning *et al.* [90] proposed an incremental-based spectral clustering approach for detecting clusters from dynamic data. The approach extends the standard spectral clustering to handle evolving data by employing *incidence matrix* to represent the dynamic data. It does so by continuously updating the eigenvalue system and generating cluster labels. Chi *et al.* [28] proposed an evolutionary spectral clustering method that incorporates temporal smoothness for detecting community structures from dynamic graphs. The method employs graph-based measures to characterize cost functions that regulate temporal smoothness in the spectral clustering to infer the corresponding optimal solutions. Kannan *et al.* [70] analyzed the performance of well-known spectral clustering for dynamic data. Also, the authors proposed a new bi-criteria metric that measures a clustering's quality, based on properties pertaining expansion of pairwise similarity graph.

### d: NONNEGATIVE MATRIX FACTORIZATION METHODS

Márquez and Weber [83] proposed a method based on non-negative matrix factorization (NMF) for detecting overlapping community structures from dynamic networks with node attributes. First, tensor's frontal slices are used for depicting the adjacency matrix at each snapshot. Then, a Bayesian approach is employed for ranking. The method employs a combination of node attributes and link information in a temporal network to strengthen the identification of communities.

Gauvin *et al.* [49] proposed a method that detects community structures from temporal networks and tracks their activities over time. The authors investigated and employed non-negative tensor factorization and latent factor decomposition techniques for extracting a community's activity structures. The method depicts a temporal network's adjacency matrix as a three-way tensor. The resulting tensor is approximated as a sum of terms interpreted as communities.

## 2) USING DIRECT UPDATE CLUSTERING CATEGORY

### a: EVOLUTIONARY CLUSTERING METHODS

Folino and Pizzuti [41] proposed a multi-objective method for detecting community structures in dynamic networks. Temporal smoothness is presented as an evolutionary multi-objective problem. The first objective is maximizing the quality of snapshots, which measures the degree of goodness of a detected community in representing the data at the current time. The second objective is minimizing temporal cost, which measures the distance between a pair of clusters at consecutive time steps.

Chakrabarti *et al.* [29] proposed an evolutionary clustering method for clustering dynamic networks over time. The authors extended traditional *k*-means algorithm to evolutionary setting. Also, the authors extended a hierarchical bottom-up agglomerative clustering algorithm to evolutionary setting. Xu *et al.* [141] proposed an evolutionary clustering method for tracking community structures in dynamic

social networks over time. The method employs an adaptive evolutionary clustering procedure. The adaptively weighted combination of historical and current data is used to detect communities at each time step.

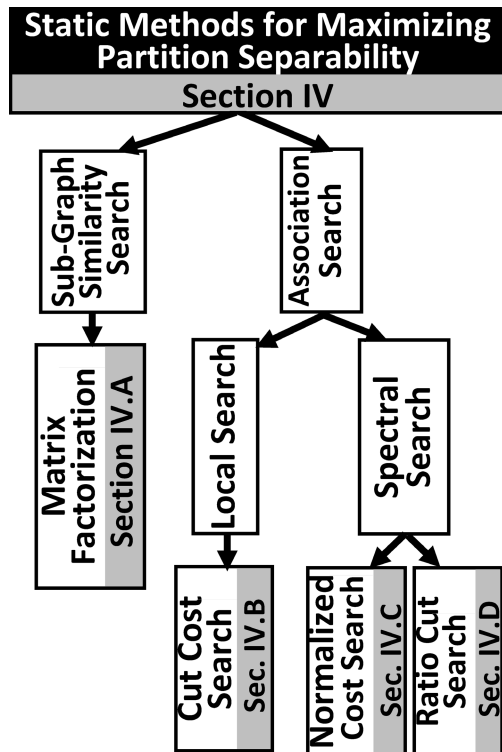
Kim and Han [69] proposed a *particle-and-density* evolutionary clustering method for detecting temporally smoothed clusters. A dynamic network is modelled as a set of lots. Community is modelled as a densely connected particles. The method employs optimal modularity and cost embedding techniques for detecting communities. Lancichinetti *et al.* [77] proposed a method that combines dominance-based strategies with a predefined resolution parameter that acts as a reference point for identifying both hierarchical structure and overlapping communities. The structure of a community is based on the peaks in the fitness histogram. The method maximizes the dynamic similarity of a community by dynamically investigating all hierarchical levels of a network based on the influences of its nodes.

Agrawal [8] proposed a bi-objective genetic method for community detection. It adaptively generates reference points and employs a non-dominated sorting approach. The method maximizes the internal density by maximizing the in-degree of a cluster's nodes, which in turn increases the modularity of clusters. Moreover, the method maximizes the dynamic similarity by constructing a cluster of competing nodes and dynamically ranks them based on their non-dominance status. Konstantinidis *et al.* [68] proposed a multi-objective evolutionary framework to search for objects/users in a mobile social community. The framework maximizes the separability of populations by employing decomposition to identify a diverse set of non-dominated objects in a single run. The method employs a priori reference point to manage a trade-off between the following two objectives: (1) maximizing the internal density of a population by increasing the recall rate of user querying, and (2) minimizing the query response time in performing a search.

### b: LABEL PROPAGATION METHODS

Xie *et al.* [140] proposed an online distributed incremental algorithm for detecting evolving communities over time in dynamic networks using stabilized label propagation. During the processing of label propagation, each node in the network employs local information only. The algorithm adopts a conditional update rule, where only the nodes involved in changes are allowed to accept the new distribution. That is, only nodes changed between consecutive snapshots are updated. This includes nodes that delete and add links. Moreover, this includes nodes removed from or added to the network.

Pang *et al.* [102] proposed an incremental label propagation method for detecting the structures of communities extracted from networks in real time. The method attempts to deal with changes in a network. It considers only locally changed nodes incrementally. First, each node will be assigned a group label number at random. It will be assigned



**FIGURE 3.** A methodology-based taxonomy, which hierarchically classifies the static community detection methods that optimize the partition separability objective function.

the label of the majority of its neighbors. Then, the label will be changed according to the labels of the neighbors.

#### IV. STATIC CLUSTERING METHODS THAT MAXIMIZE THE PARTITION SEPARABILITY

These methods employ the *underlying techniques* of the network-centric category by ensuring that each two connected vertices within a community are closely associated. They consider network-based properties such as the ratio of the number of edges connecting a vertex within and outside a community and the position of a vertex in the latent space. We review and discuss in this section the static clustering methods that optimize the partition separability objective function. Fig. 3 presents our methodology-based taxonomy, which hierarchically classifies the static community detection methods that optimize the partition separability objective function.

##### A. METHODS THAT APPLY MATRIX-BASED EIGENVECTORS AND FACTORIZATION SUB-GRAPH SIMILARITY SEARCH

Most of these methods consider that magnitudes represent a good measure of the degree of belonging (i.e., strength) of vertices to communities. Some of them use centrality index to quantify the degree of vertices' influences in a community. They compute some type of centrality indexes based on the magnitudes of the elements of the eigenvector of the matrix under consideration. Ye *et al.* [144] proposed a nonnegative

matrix factorization method for detecting discrete overlapping communities. The discrete community membership of each node is determined directly without the need of post-processing. The method employs a combination of kernel regression and discriminative pseudo supervision strategies. Newman [86] proposed an algorithm for identifying the community structure in a network using eigenvectors of modularity matrix. The algorithm employs modularity objective function to detect the hierarchical structure of clusters. The algorithm divides a network into two parts, if the modularity rises above a certain threshold. The division is done based on leading Eigen vector in the modularity matrix.

##### B. METHODS THAT APPLY CUT COST LOCAL ASSOCIATION SEARCH

These methods simplify the process of finding cuts to minimize and make the cost of computing approximations more flexible. Veldt *et al.* [129] proposed an approximation clustering method based on the objective of sparsest cut's multiplicative scaling and weighted correlation clustering. The method combines sparsest cut and other quality functions. It selects a node at random iteratively. It builds a cluster by greedily aggregating nodes adjacent to the selected node. Kernighan and Lin [62] proposed an algorithm that minimizes the difference between the number of inter-community links and intra-community links. The algorithm swaps or moves vertices between communities iteratively for the sake of decreasing the evaluation function. This process terminates when the evaluation function becomes unchanged. Andersen *et al.* [2] proposed a local partitioning method that simplifies the process of identifying cuts. The method employs a single approximate PageRank vector instead of a sequence of random walk vectors. This also makes the cost of computing approximations more flexible. Karypis and Kumar [64] proposed a graph coarsening heuristic algorithm. The algorithm allows the size of coarse graph partitioning to be small relative to the overall size of the final partitioning, which is determined after multilevel refinements.

##### C. METHODS THAT APPLY NORMALIZED COST SPECTRAL ASSOCIATION SEARCH

Spectral methods employ quadratic optimization techniques to optimize some pre-defined cut criteria. The cut criteria for the bipartition of a network is the number of inter-group links. It is considered optimal, if it produces minimum cut. However, this minimum cut criterion can result in bias partitions. To overcome this, a number of methods proposed other criteria. For example, some methods attempt to approximate the optimal cut by transforming it into a constraint quadratic optimization problem. Shi and Malik [115] proposed a normalized cut method to compute the density of a partition rather than the number of inter-group links inside the partition. It is a variant of the Laplacian-based matrix method. The optimal solution is achieved by calculating the second smallest eigenvector of the symmetric positive semi-definite matrix.

#### D. METHODS THAT APPLY RATIO CUT SPECTRAL ASSOCIATION SEARCH

These methods assume that a given vertex is likely to belong to a certain partition, if the number of edges linking this vertex with vertices inside the partition is the same or higher than the number of edges linking it with vertices that are part of other partitions. Wei and Cheng [131] proposed a ratio cut method for identifying the clustering structures for hierarchical Very-Large-Scale Integration (VLSI). The method solves the ratio cut problem via multi-commodity flow formulation using linear programming techniques. Flake *et al.* [35] proposed a method based on the graph theory's Max Flow-Min Cut theorem. The authors assume that the maximum flow within a network can be determined by the capacity of the minimum cut sets. They considered sparse inter-community links as "bottlenecks" in the flow. Therefore, they identify inter-community links through minimum cut sets computation by iteratively removing "bottleneck" links.

#### V. STATIC METHODS THAT MAXIMIZE THE DYNAMIC SIMILARITY

We review and discuss in this section the static clustering methods that optimize the dynamic similarity objective function. Fig. 4 presents our methodology-based taxonomy, which hierarchically classifies the static community detection methods that optimize the dynamic similarity objective function.

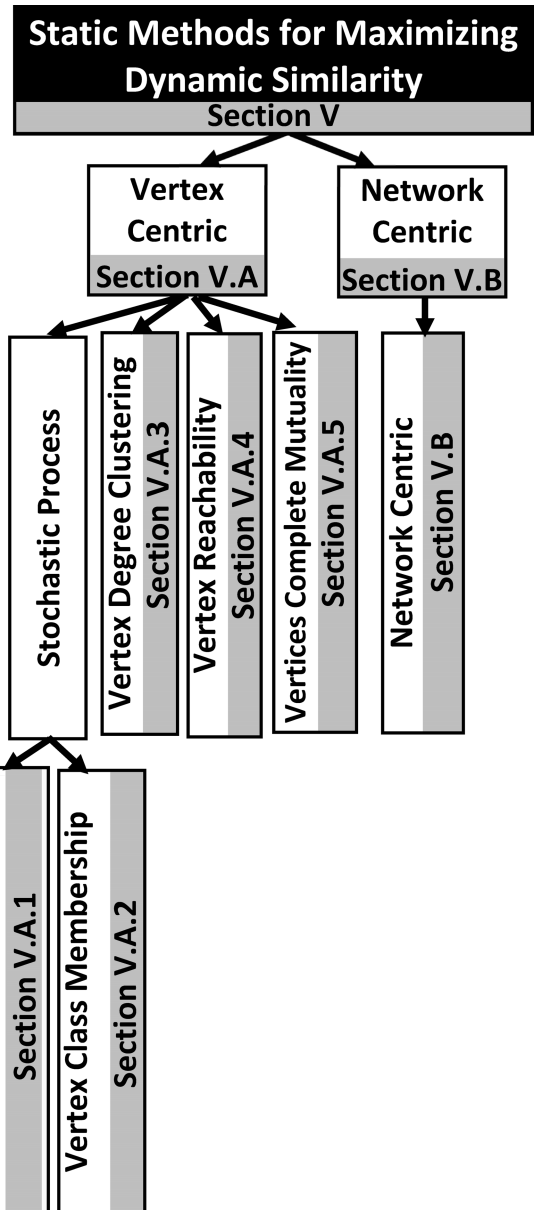
##### A. VERTEX CENTRIC CLUSTERING CATEGORY

The methods of this category fall under the following five fine-grained classes: vertex random walk distance-based (Section V.A.1), vertex class membership-based (Section V.A.2), vertex degree-based (Section V.A.3), vertex reachability-based (Section V.A.4), and vertices complete mutuality-based (Section V.A.5).

##### 1) METHODS THAT APPLY VERTEX RANDOM WALK DISTANCE BASED ON STOCHASTIC PROCESS

These methods aim at combining the accuracy of global processing with the efficiency of local search. They employ random walk to gain knowledge about the network's topology. This helps in the local heuristic search that detect partitions. Meng *et al.* [80] proposed a  $K$ -path initialization-based method for detecting community structures in complex social networks by analyzing their topology structures' information. The probability of a node to be selected increases as its degree increases. A  $k$  walk length is an indicative that the number of walk attempts at a given time is  $k$ . The input to the algorithm is a graph ( $G = V, E$ ), the number of walks, the maximum length of walk path  $k$ , and the uninitialized nodes. The output is a set of initialized nodes.

As a global preprocessing, the Complex Network Cluster Detection method proposed by De Meo *et al.* [30] computes the  $k$ -path centrality of each edge using the ERW-Kpath



**FIGURE 4.** A methodology-based taxonomy, which hierarchically classifies the static community detection methods that optimize the dynamic similarity objective function.

algorithm [30], which approximates the centrality of the edge by calculating its probability of being part of random non-backtracking walks of length  $k$ . Then, the distances between all pairs of vertices are computed using the  $k$ -path centralities of the edges connecting them. Each edge is assigned a weight that corresponds to the distance between the vertices at the end points of the edge. The Louvain Method proposed by Blondel *et al.* [12] clusters a network based on the weights of its edges. As a global preprocessing, the Infomap method [107] computes the shortest description lengths of random walks based on concepts of information theory. The description length is defined as the number of

bits that a vertex needs to encode the random walk's path. It is computed using map equation [109].

## 2) METHODS THAT APPLY VERTEX CLASS MEMBERSHIP BASED ON STOCHASTIC PROCESS

These methods divide a network by selecting edges and the vertices at their endpoints uniformly at random using some probability distribution. They cluster them accordingly. Cui *et al.* [24] proposed a null network-based method that applies vertex and edge class membership for detecting communities. Only a community's inner structure is changed. However, the characteristics of community's structure and the number of communities are maintained without change. The proposed method maintains the following two types of link relationships: edges between communities and edges within a community.

Palowitch *et al.* [100] proposed a method for detecting communities from weighted networks. It applies vertex class membership using iterative hypothesis tests under iterative explicit null model. This includes sequential significance tests. The method adaptively detects communities sharing nodes. It ignores nodes that are insignificantly linked to any community. The method proposed by Pizzuti [96] selects a vertex  $v$  and one of its neighbors  $u$  at random. It creates an initial division of the network from the pair  $v$  and  $u$ . The method proposed by Liu *et al.* [71] selects the vertices of a division by employing Markov random walk. These vertices should satisfy Markov chains, where the marginal and conditional distributions are multivariate normal.

Watts [132] proposed a mixed membership model that transforms multivariate normal distributions (probabilities of class membership and ties between various classes) into domain of probability vectors. Erdos and Renyi [34] proposed a model that selects vertices connected by edges at random. The probability of two vertices to be connected by an edge is  $p$ ; otherwise, it is  $1 - p$ .

## 3) METHODS THAT CLUSTER BASED ON THE DEGREES OF VERTICES

These methods cluster a network in such a way that each vertex within a partition is adjacent to a large number of vertices confined within the partition [9]. They classify partitions into two substructures called  $k$ -core and  $k$ -plex. A substructure  $k$ -plex is a subnetwork with  $m$  vertices, each of them is adjacent to at least  $m-k$  vertices in the subnetwork. When  $k = 1$ , a  $k$ -plex becomes a clique. A substructure  $k$ -core is a subnetwork, where each vertex is connected to at least  $k$  vertices in the subnetwork.

## 4) METHODS THAT CLUSTER BASED ON THE REACHABILITY OF VERTICES

These methods determine a partition based on the reachability between its vertices [63]. Thus, there should be a short path between any two vertices in a partition.

## 5) METHODS THAT CONSIDER VERTICES COMPLETE MUTUALITY

In these methods, a partition is a maximal complete sub-network of vertices adjacent to each other (i.e., a clique) [67].

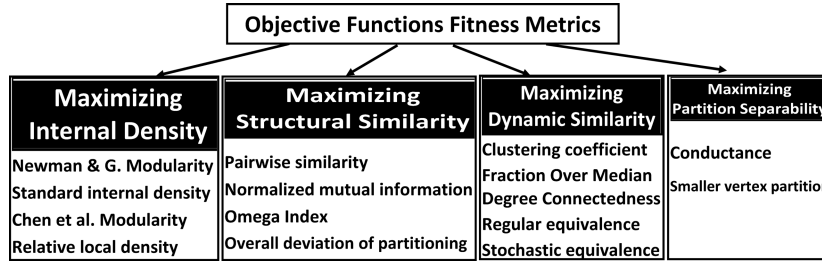
## B. NETWORK CENTRIC CLUSTERING CATEGORY

The methods that fall under this category do not define a community independently. They define a community based on the topology of the entire network. They consider network-based properties such as the ratio of the number of edges connecting a vertex within and outside a community and the position of a vertex in the latent space. The vertex-influence methods fall under this category. These methods measure the global relative influence of each vertex in controlling the flow of information in a network. They consider the task of identifying the influential individuals in a social network as a discrete optimization problem. They infer the social status of individuals based on the influences of the vertices representing them in controlling the flow of information in the network [53], [119]. Most of these methods are based on the notion that a set of vertices influenced by a same influential vertex  $v$  must have rather a similar interaction pattern with  $v$  [1], [4], [61], [73], [119].

Kempe *et al.* [61] proposed a framework based on submodular functions. The authors considered the issue of selecting the set of influential vertices in a network as a discrete optimization problem. They employed operational models from mathematical sociology to identify a joint distribution over the behavior of all vertices in a network. A number of methods that adopt different techniques have been proposed for identifying individuals, who share a same social position [59], [119] or individuals, who are influential in a social network [120], [121], [136]. Jiang *et al.* [59] proposed the concepts of "abnormal" and "synchronized" vertices to identify the patterns of behavior of vertices representing criminals in a criminal network. These concepts lead to the identification of suspicious vertices. Taha and Yoo [121], [126] proposed two forensic analysis systems that can identify the influential members and the immediate leaders of lower-level criminals in a criminal organization. In the first system, they employed the concept of existence dependency. Each vertex is given a score that represents the number of other vertices, whose existence in the Minimum Spanning Tree of the network is dependent on this vertex. Vertices are ranked based their scores. The top-ranked ones represent the influential criminals. In the second system, they employed formulas that quantify the degree of influence of each vertex relative to the other vertices in a network depicting a criminal organization.

## VI. FITNESS METRICS FOR CLUSTERING OBJECTIVE FUNCTIONS

We present in this section the fitness metrics for evaluating the goodness of clustering algorithms for satisfying the internal density objective function (Section VI.A), structural simi-



**FIGURE 5.** The fitness metrics for evaluating the goodness of clustering algorithms for satisfying each clustering objective function.

larity objective function (Section VI.B), dynamic similarity objective function (Section VI.C), and partition separability objective function (Section VI.D). Fig. 5 lists the set of fitness metrics for each objective function.

#### A. FINESSES METRICS FOR THE INTERNAL DENSITY OBJECTIVE FUNCTION

The *standard internal density* metric [58] is a metric of concentration. It identifies the essential qualities of a good cluster as compact and tightly connected internally, while well-separated from other clusters. Most internal density metrics define internal density as the ratio of the: (1) number of edges connecting a partition's vertices, and (2) number of edges that have at least one endpoint confined within the partition. In our evaluations, we used the standard internal density metric [40], which is defined in Equation 1.

$$f(S) = \frac{m_s}{n_s(n_s - 1)/2} \quad (1)$$

where  $m_s$  is the number of edges within partition  $S$  and  $n_s$  is the number of vertices within partition  $S$ .

The *Chen et al. modularity* metric [19] is a widely used quality function based on modularity. It measures how well a network is divided into modules. It is based on the notion that a random network is not expected to have a cluster structure. We used this metric in our evaluation. We also used in our evaluations the popular *modularity* metric proposed by Chen et al. [21]. Let  $F$  be the fraction of edges that connect the vertices within a module  $S$ . The metric is based on the following notion. If edges are constructed at random and if  $F$  is larger than expected, then there are many more edges inside  $S$  than one can expect by random means. The *Newman-Girvan modularity* metric [87] is one of the widely used clustering metrics in literature. It is a global quality metric.

*Omega Index* [26] is a good metric for evaluating the cohesiveness of clusters based on their connections. Since cohesiveness can characterize the internal density of a cluster, we use Omega Index as a metric for evaluating the degree of concentration of clusters. It employs Rand Index for correcting chance agreements. It takes into consideration the number of clusters that contain the occurrences of same pairs of nodes.

#### B. FINESSES METRIC FOR THE STRUCTURAL SIMILARITY OBJECTIVE FUNCTION

The *pairwise similarity* metric measures the similarity between a community detected by an algorithm and some ground-truth community. Let  $C_i(v)$  be the set of vertices that appear with vertex  $v$  in community  $i$ . The pairwise similarity between two communities  $A$  and  $B$  is given by Equation 2.

$$S(A, B) = \frac{1}{n} \sum_{v \in V} \frac{|C_A(v) \cap C_B(v)|}{|C_A(v) \cup C_B(v)|} \quad (2)$$

*Mutual Information*  $I(A, B)$  measures the information of the membership of all vertex-pairs in a community  $A$  compared to a community  $B$ , and vice versa. This is defined as  $I(A, B) = H(A) - H(A|B)$ , where  $H(A)$  is the Shannon entropy, which is defined in Equation 3:

$$H(A) = - \sum_k P(a_k) \log P(a_k) \quad (3)$$

where  $P(a_k)$  and  $P(b_k)$  are the probabilities of a random vertex in the  $k^{\text{th}}$  partitions  $A$  and  $B$ , respectively.

*Normalized Mutual Information (NMI)* is the normalization of Mutual Information by averaging the entropies  $H(A)$  and  $H(B)$  as defined in Equation 4.

$$NMI(A, B) = \frac{I(A, B)}{[H(A) + H(B)]/2} \quad (4)$$

#### C. FINESSES METRIC FOR THE DYNAMIC SIMILARITY OBJECTIVE FUNCTION

The *clustering coefficient* metric measures transitivity, which is an important local property of networks [130]. It reflects the cohesion level between the neighbors of a vertex. The clustering coefficient  $c_v$  of vertex  $v$  is computed as the ratio between: (1) the number of edges linking the neighbors of  $v$ , and (2) the total number of edges. The *regular equivalence* metric identifies vertices that serve similar structural roles in terms of their connectivity profiles (e.g., reachability and adjacency). It assumes that if some vertices behave in the same role during interaction, the social status of the individuals represented by the vertices is likely to be similar.

#### D. FINESSES METRIC FOR THE PARTITIONS SEPARABILITY OBJECTIVE FUNCTION

*Conductance* metric measures the goodness of a partition in terms of how well evenly connected internally. In other

**TABLE 1.** Description of the ground-truth communities.

Name	Number of Vertices	Number of Edges	Number of Communities
com-LiveJournal	3,997,962	34,681,189	287,512
com-Friendster	65,608,366	1,806,067,135	957,154
com-Orkut	3,072,441	117,185,083	6,288,363
com-DBLP	317,080	1,049,866	13,477
com-Amazon	334,863	925,872	75,149

words, it measures how hard to split a partition into two sub partitions. Conductance of the internal cut characterizes this notion. Formally, given a network  $N = (V, E)$  and a set  $S \subseteq V$ , conductance is defined as shown in Equation 5:

$$\frac{Q_S}{2E + Q_S} \quad (5)$$

- $E_S = |\{(u, v) \in E | u, v \in S\}|$ : the number of edges in  $S$ .
- $Q_S = |\{(u, v) | u \in S, v \notin S\}|$ : the number of edges connecting the vertices in  $S$  with other vertices outside  $S$ .

## VII. EVALUATING THE STATIC COMMUNITY DETECTION METHODS EMPIRICALLY

Detected communities can be labelled as “bad” or “good” based on widely known and agreed upon properties that capture the notion of quality. These properties are known in literature as clustering objective functions. In this section, we empirically compare the different methods that optimize each of the following four objective functions: internal density, structural similarity, dynamic similarity, and partitions separability objective functions. We used the fitness metrics for the objective functions described in Section VI for the evaluations. From the set of fitness metrics for each clustering objective function, we selected the most popular and widely used ones.

### A. COMPILING GROUND TRUTH COMMUNITIES

We used five sets of ground-truth communities compiled by the Stanford Network Analysis Project (SNAP) [78]. These ground-truth communities allowed us to evaluate the methods quantitatively. The five ground-truth datasets are listed in Table 1.

### B. EVALUATION SETUP

We ran the prototypes adopting the different methods using Windows 10 Pro and Intel(R) Core(TM) i7-6820HQ processor. The CPU and RAM of the machine have 2.70 GHz and 16 GB, respectively. We performed the following procedure for the empirical evaluations:

- 1) For each of the static clustering methods described in Sections II-V, we selected one of the proposed techniques that falls under the method (i.e., adopts the underlying principles of the method). That is, for each method, we selected a paper, whose proposed technique employs the underlying principles of the method. We considered the selected technique/paper

as a representative of the method. From among all papers proposed techniques adopting one of the methods, we selected the most influential one. We based the influence of a paper on factors such as its number of citations, recency, and state of the art. We evaluated each selected technique using the different fitness metrics for the objective function, which the technique attempts to optimize (recall Section VI and Fig. 5).

- 2) For each clustering category, we ranked the different methods that fall under the category. The ranking was performed by averaging the fitness scores achieved by each technique representing a method.
- 3) For each clustering objective function, we ranked the different clustering categories that fall under the objective function. The ranking was performed by averaging the fitness scores achieved by the methods that fall under each clustering category.

Thus, based on the fitness scores achieved by the selected techniques, we ranked the different methods and clustering categories. We also ranked the different clustering categories that fall under each objective function.

### C. EVALUATING THE METHODS THAT MAXIMIZE THE INTERNAL DENSITY OBJECTIVE FUNCTION

In this test, we use the following metrics described in Section VI for measuring the accuracies of the methods that maximize the internal density objective function: (1) Newman & G. modularity [87], (2) Chen *et al.* modularity [19], and (3) standard internal density metrics [58]. These metrics measures how well a network is divided into modules. A network is identified as highly modular, if the vertices within its modules are densely connected and it has sparse inter-modules connections. The *Chen et al. modularity* metric is based on the notion that a random network is not expected to have a cluster structure. The Newman-Girvan modularity metric is a global quality metric and it measures how well edges are clustered within detected communities. Table 2 shows the results.

### D. EVALUATING THE METHODS THAT MAXIMIZE THE STRUCTURAL SIMILARITY OBJECTIVE FUNCTION

In this test, we use the following metrics described in Section VI for measuring the accuracies of the methods that maximize the structural similarity objective function: (1) Normalized Mutual Information (NMI) (recall Equation 4), (2) pairwise similarity (recall Equation 2), and (3) Omega Index [26]. NMI measures the information of the membership of all vertex-pairs in a detected community compared to a ground truth community. It measures the distance between the clustering result of a method and the ground truth at each time. The value of NMI ranges from 0 to 1 and a higher the value the better accuracy. Pairwise similarity metric measures the proximity between the vertices of communities based on their structural similarities. The Omega Index

**TABLE 2.** The Fitness score of each technique representing a method, the ranking of the different methods that fall under a same clustering category, and the ranking of the different clustering categories that fall under a same objective function.

Objective function	Clustering category	Clustering method	Rep. tech.	Fitness metrics for the objective functions	Tech. score	Method rank	Category rank
Maximizing Internal Density	Hierarchy Centric	Top-down divisive method	[89]	Newman & G. modularity	0.63	1	1
				Standard internal density	0.58		
				Chen et al. modularity	0.53		
		Bottom-up agglomerative method	[14]	Newman & G. modularity	0.54	2	
				Standard internal density	0.61		
				Chen et al. modularity	0.46		
	Bottom-up intermediary score maximization method	[88]	Newman & G. modularity	0.51	3		
			Standard internal density	0.53			
			Chen et al. modularity	0.43			
	Group Centric	Density centric method	[84]	Newman & G. modularity	0.47	4	2
				Standard internal density	0.54		
				Chen et al. modularity	0.38		
Vertex Centric	Embedding-based method	[46]	Newman & G. modularity	0.42	5	3	
			Standard internal density	0.53			
			Chen et al. modularity	0.35			
Maximizing Structural Similarity	Network Centric	Block-based sub-graph similarity heuristic search method	[66]	Pairwise similarity	0.88	1	1
				Normalized mutual information	0.82		
				Omega index	0.71		
		Clique-based sub-graph similarity heuristic search method	[143]	Pairwise similarity	0.84	2	
				Normalized mutual information	0.78		
				Omega index	0.66		
	Latent space local association search method	[116]	Pairwise similarity	0.85	4		
			Normalized mutual information	0.67			
			Omega index	0.54			
	Vertex Centric	Label propagation based on stochastic process method	[74]	Pairwise similarity	0.87	3	2
				Normalized mutual information	0.74		
				Omega index	0.58		
Vertices community variance method		[111]	Pairwise similarity	0.72	5		
			Normalized mutual information	0.58			
			Omega index	0.54			
Group Centric	Locality sensitive hashing method	[78]	Pairwise similarity	0.65	6	3	
			Normalized mutual information	0.57			
			Omega index	0.45			
Maximizing Dynamic Similarity	Network Centric	Vertex influence-heuristic method	[61]	Clustering coefficient	0.63	1	1
				Fraction Over Median Degree	0.59		
	Vertex Centric	Vertex random walk distance method	[140]	Clustering coefficient	0.57	2	2
				Fraction Over Median Degree	0.53		
		Vertex class membership method	[68]	Clustering coefficient	0.46	6	
				Fraction Over Median Degree	0.43		
		Vertex degree method	[9]	Clustering coefficient	0.52	4	
				Fraction Over Median Degree	0.51		
	Vertices complete mutuality method	[67]	Clustering coefficient	0.54	5		
			Fraction Over Median Degree	0.46			
	Vertex reachability	[63]	Clustering coefficient	0.57	3		
			Fraction Over Median Degree	0.49			
Maximizing Partitions Separability	Network Centric	Matrix eigenvectors	[144]	Conductance	0.66	1	N/A
		Normalized cost search	[129]	Conductance	0.57	2	
		Ratio cut search	[131]	Conductance	0.52	3	
		Cut cost search	[62]	Conductance	0.43	4	

"Rep." and "tech." are abbreviations of the terms "representative" and "technique" respectively

considers the number of clusters that contain the occurrences of same pairs of nodes. Table 2 shows the results.

#### **E. EVALUATING THE METHODS THAT MAXIMIZE THE DYNAMIC SIMILARITY OBJECTIVE FUNCTION**

In this test, we use the following metrics described in Section VI for measuring the accuracies of the methods that maximize the dynamic similarity objective function: (1) clustering coefficient, and (2) Fraction Over Median Degree. Clustering coefficient is based on the notion that a community is an indication of inhomogeneous distributions of edges; therefore, vertex pairs that have common neighbors are likely to be linked with each other [130]. Fraction Over Median Degree considers vertices are similar, if they share same connection pattern [33], [53]. Table 2 shows the results.

#### **F. EVALUATING THE METHODS THAT MAXIMIZE THE PARTITIONS SEPARABILITY OBJECTIVE FUNCTION**

In this test, we use the conductance metric described in Section VI (recall Equation 5) for measuring the accuracies of the methods that maximize the partitions separability objective function. Conductance measures how well a detected community connected evenly internally. Table 2 shows the results.

#### **G. DISCUSSION OF THE RESULTS**

Table 2 shows the fitness score of each technique representing a method, the ranking of the different methods that fall under a same clustering category, and the ranking of the different clustering categories that fall under a same objective function. In each of the next Subsections 1-4, we discuss our observation of the results of the methods and clustering categories that maximize each objective function.

##### **1) DISCUSSION OF THE RESULTS OF THE METHODS AND CATEGORIES THAT MAXIMIZE THE INTERNAL DENSITY OBJECTIVE FUNCTION**

Among the clustering categories that maximize the internal density objective function, the results of the empirical evaluations revealed that the hierarchy centric category achieved better results than the vertex centric and group centric categories. This is because, based on the results, the vertex centric and group centric categories worked well only in specific types of networks and problems. Specifically, they worked well only in:

- 1) bisection networks,
- 2) networks with lot of vertices that serve similar structural roles in terms of their connectivity profiles, and/or
- 3) networks with lots of vertices that have strong connectivity.

The vertex centric and group centric categories performed poorly in more general problems, especially those required the entire network topology or/and the interaction roles of vertices to be analyzed. The hierarchy centric category combated these limitations by considering the topology of the

entire network and by employing the betweenness centralities of edges for analyzing the interaction patterns of vertices to identify the boundaries of communities. This is because community structure detection is largely a nonlocal problem.

Among the methods that fall under the hierarchy centric category, the top-down divisive method achieved better results than the bottom-up agglomerative and bottom-up intermediary score maximization methods. This is because the divisive method optimized modularity over all possible divisions to identify the best one. Instead of determining which edges are most central to a community, the divisive method focused on identifying the least central ones. The method constructed communities by progressively removing the least central edges. The results of the evaluations revealed that the agglomerative method has a limitation caused by its tendency to identify only the core vertices of communities and overlooks the peripheral ones. Core vertices have strong similarities; therefore, they are connected early in the agglomerative procedure. Peripheral vertices that have weaker similarities to other vertices tend to be overlooked. The bottom-up agglomerative method suffered from a resolution limit, where communities below the threshold needed to be merged. They did not guarantee optimal network partitions.

##### **2) DISCUSSION OF THE RESULTS OF THE METHODS AND CATEGORIES THAT MAXIMIZE THE STRUCTURAL SIMILARITY OBJECTIVE FUNCTION**

Among the clustering categories that maximize the structural similarity objective function, the results of the empirical evaluations revealed that the network centric category was significantly more general than the vertex centric and group centric categories. This is due, in part, to the fact that the network centric category could detect many forms of community structures as well as simple communities of dense links. Unlike the vertex centric and group centric categories, the network centric category showed the desirable property of asymptotic consistency under certain conditions. Moreover, the network centric category considered successfully the heterogeneity in the degrees of vertices. It considered the degree distribution in the networks by analyzing various types of degree heterogeneity, which improved the fitting to the networks' data. This eventually led to better community detection accuracy.

Among the methods that fall under the network centric category, the clique-based sub-graph similarity heuristic search method achieved better results than the block-based sub-graph similarity heuristic search and the latent space local association search methods. This is due, in part, to the inclusion of the degree-corrected model in the clique-based sub-graph similarity method for inferring group structure proved to produce more accurate community structures than the uncorrected model. An uncorrected model tended to split a network with a substantial degree of heterogeneity into groups of low and high degree. This tendency prevented such a model from detecting correct group memberships.



The degree-corrected model adopted by the clique-based sub-graph similarity method correctly overlooked divisions based only on degree of heterogeneity. Therefore, it is more suitable to underlying structures. However, the degree-corrected model showed some limitations. The model returned sometimes unrealistic number of zero-degree vertices. It was also unable sometimes to deal with some degree sequences. Sometimes, it represented higher-order network structure incorrectly. Moreover, the model assumed that the number  $k$  of blocks was given, which is a limitation.

### 3) DISCUSSION OF THE RESULTS OF THE METHODS AND CATEGORIES THAT MAXIMIZE THE DYNAMIC SIMILARITY OBJECTIVE FUNCTION

Among the clustering categories that maximize the dynamic similarity objective function, the results of the empirical evaluations revealed that the network centric category achieved better results than the vertex centric category. Specifically, the natural greedy strategy adopted by the network centric category outperformed the high-degree, centrality, and random heuristic techniques adopted by the following vertex centric methods:

- 1) Vertex random walk distance based on stochastic process.
- 2) Vertex reachability.
- 3) Vertex degree.
- 4) Vertices complete mutuality.
- 5) Vertex class membership.

The evaluation results revealed that vertex centric category overlooked the fact that some of the highest-degree or/and central vertices may be clustered. As a result, the clustering consideration of all these vertices may not be unnecessary. Actually, the global influences and behaviors of vertices may not be accurately reflected by the centralities and degrees of these vertices. Initially, the methods that employ degree and centrality techniques identified influential vertices better than the methods that employ random heuristic techniques. However, eventually the methods that employed random heuristic techniques surpassed the methods that employed degree and centrality techniques in identifying all influential vertices because they did not focus exclusively on central vertices.

The vertex influence-based heuristic method performed joint distribution over the behavior of all vertices globally, which positively contributed to the method's performance. The results revealed that the distance-based and random-based methods had the limitation of clustering most seed vertices, because the distances between vertices in large communities are usually small. The seed-based clustering technique adopted by the vertex centric methods may significantly affect the identification of influential vertices. This is due to the impact of influence spread. Overall, the results showed that the network centric category had the advantage of considering the dynamicity of information instead of relying only on the structural relationships between vertices.

### 4) DISCUSSION OF THE RESULTS OF THE METHODS AND CATEGORIES THAT MAXIMIZE THE PARTITION SEPARABILITY OBJECTIVE FUNCTION

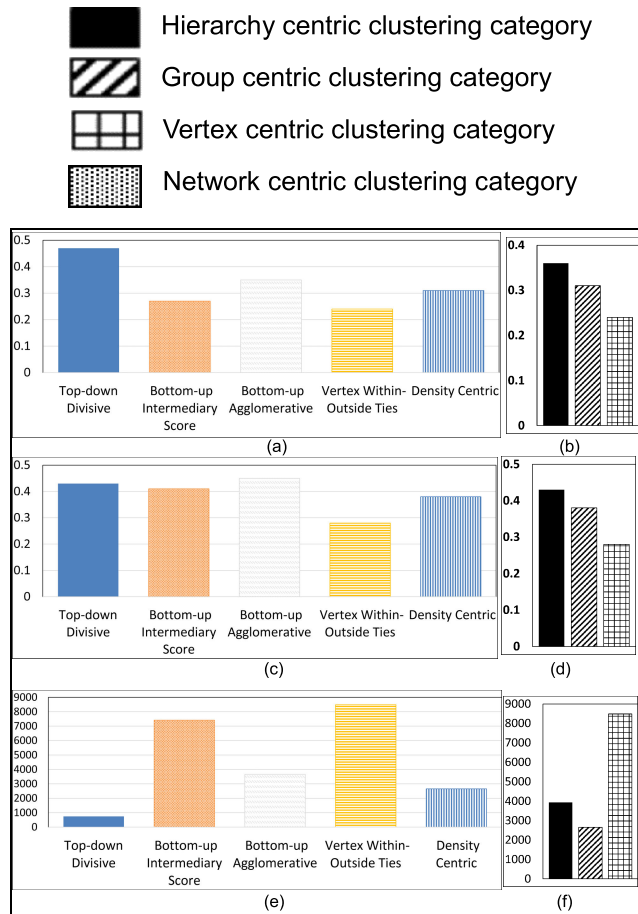
Within the network centric category, the results of the empirical evaluations revealed that the matrix-based eigenvectors method outperformed the normalized cost search, ratio cut search, and cut cost search methods. This is due, in part, to the employment of the matrix-based eigenvectors to global optimal criteria for performing segmentation instead of focusing only on local features and network topology. The cut cost search and ratio cut search methods identify local (as opposed to global) optimal solutions. The two methods accepted only better neighbor solutions during local search process and disregard worse ones. They required prior knowledge about the average size and number of communities for detecting an initial partition. Predicting prior knowledge values is not as easy task. Also, the two methods relied on the accuracy of the initially detected partitions. Inaccurate initial partitions may cause a slow convergence. As a result, the two methods produced several poor solutions. The cut cost search method tended to produce small cut sets of isolated vertices. That is, the method is bias towards partitioning small sets of vertices.

## VIII. EVALUATING THE STATIC COMMUNITY DETECTION METHODS EXPERIMENTALLY

In this section, we experimentally compare the different static clustering methods described in Sections II-V. We followed the same procedure described in Subsection VII.B for selecting the techniques that represent the different methods. That is, for each method, we selected a technique/paper as a representative of the method. From among all papers that proposed techniques adopting one of the methods, we selected the most influential one (recall Subsection VII.B for more details). We also used the five sets of ground-truth communities described in Subsection VII.A and listed in Table 1. We evaluated the accuracy of the methods for detecting communities from scratch (Section VIII.A) and for assigning new members to existing communities (Section VIII.B).

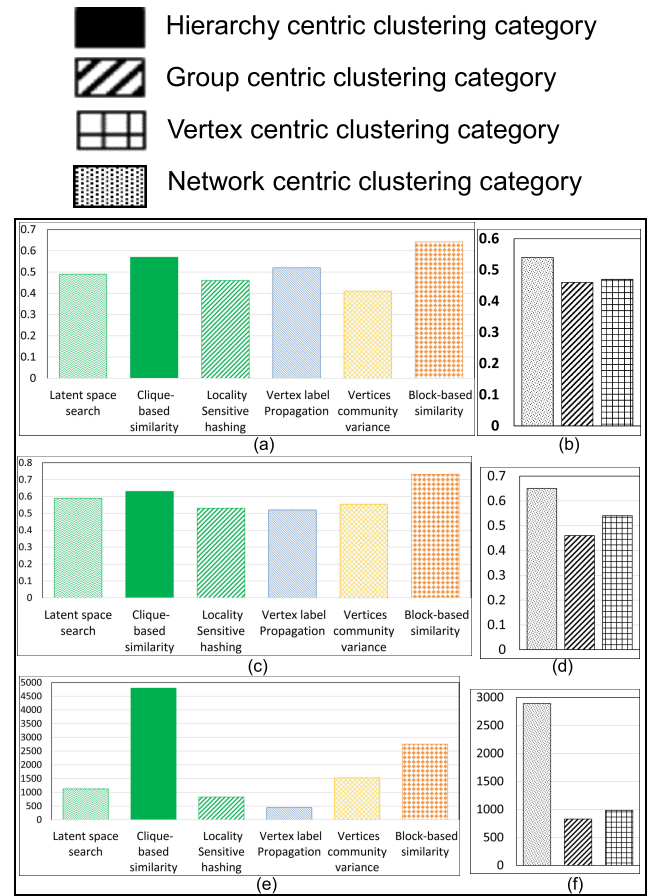
### A. EVALUATING THE ACCURACY OF DETECTING COMMUNITIES FROM SCRATCH

In this test, we compared the accuracies of the different methods and clustering categories for detecting the ground-truth communities listed in Table 1 from scratch (as opposed to augmenting existing partial communities). We computed the Adjusted Rand Index (ARI) and F1-score for each method by comparing its predicted communities with the corresponding ground truth communities shown in Table 1. That is, we computed the ARIs and F1-scores with reference to the actual ground-truth communities. Figs. 6-9 plot the results of comparing the methods and clustering categories in terms of their ARIs and F1-scores for maximizing the internal density, structural similarity, dynamic similarity, and partition separability objective functions, respectively.



**FIGURE 6.** Comparing the clustering methods and categories that maximize the internal density objective function, where: (a) average F1-score of the clustering methods, (b) average F1-score of the clustering categories, (c) average ARI of the clustering methods, (d) average ARI of the clustering categories, (e) average execution time (in seconds) of the clustering methods, and (f) average execution time (in seconds) of the clustering categories.

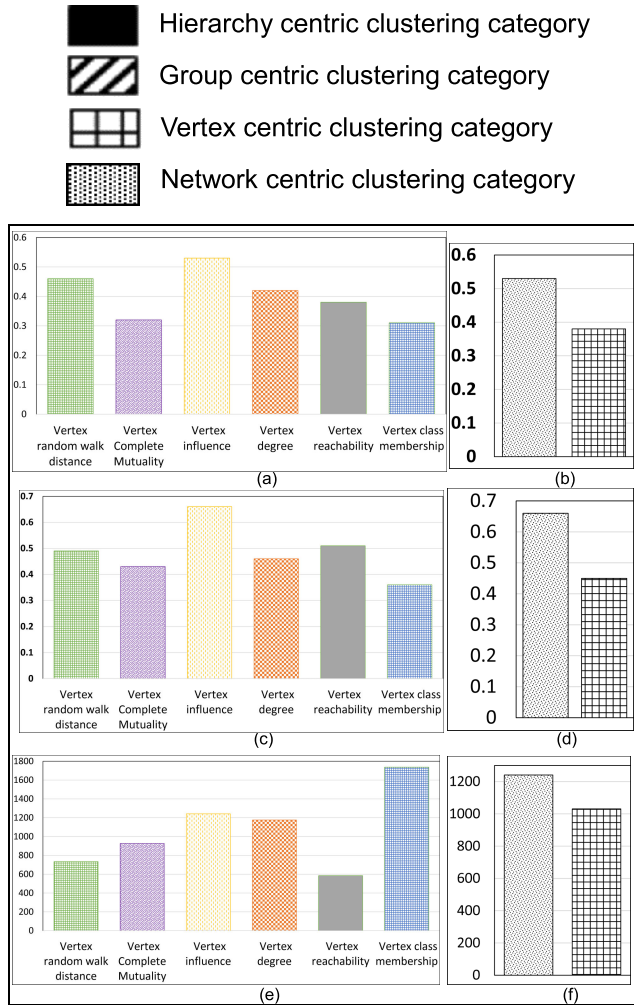
As the results of the methods and categories that optimize the internal density objective function plotted in Fig. 6 show, the hierarchy centric clustering category (represented by the top-down divisive, bottom-up intermediary score, and bottom-up agglomerative methods) outperformed the group centric clustering category (represented by the vertex within-outside ties method) and the vertex centric clustering category (represented by the density centric method). Based on our observation of the results, we attributed the good performance of the hierarchy centric category to its employment of global preprocessing techniques that consider the topology of the entire network. The category predicted each vertex-vertex degree of association according to global vertex-edge associations. It predicted the association between each pair of vertices  $v_i$  and  $v_j$  based on the associations between the edges in the path from  $v_i$  to  $v_j$  and each of  $v_i$  and  $v_j$ . In other words, the degrees of association between vertices are based on the degrees of influences of the edges connecting them. The group centric and group centric categories detected communities in independence of how closely



**FIGURE 7.** Comparing the clustering methods and categories that maximize the structural similarity objective function, where: (a) average F1-score of the clustering methods, (b) average F1-score of the clustering categories, (c) average ARI of the clustering methods, (d) avg ARI of the clustering categories, (e) avg execution time (seconds) of the clustering methods, and (f) avg execution time of the clustering categories.

associated their connections are based on the global influences of the edges connecting them.

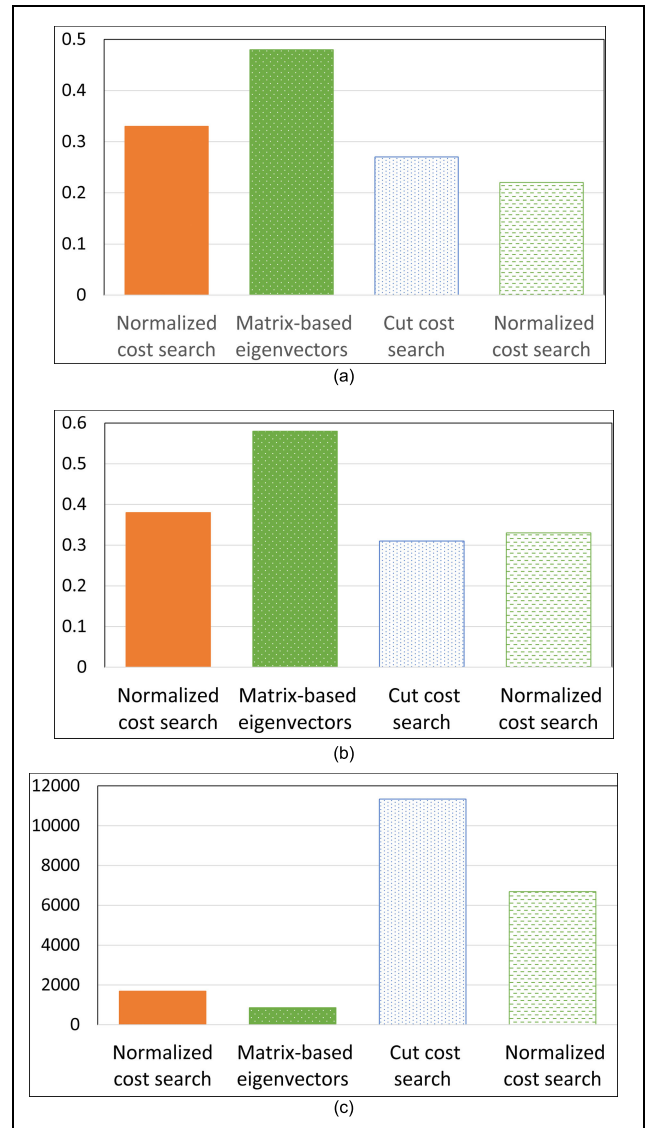
Due to the specificity nature of the structural characteristics of a vertex in a network, the network centric category outperformed the group centric and vertex centric categories by considering the structural similarities of vertices for optimizing the structural similarity objective function as shown in Fig. 7. Based on our analysis of the experimental results, we observed that the structural-based search properties required for identifying communities with certain structural similarities are included within the underlying techniques of the network centric methods. Specially, the network centric methods considered the following: (1) the ratio of the number of edges connecting a vertex within and outside a community, and (2) the position in the latent space of a vertex. Thus, the network centric methods considered the connections in the entire network, where a community was not defined independently. This ensured that each pair of vertices within a community was connected by edges that are closely associated with the other vertices in the community. The degrees of association between vertices and the edges connecting them were based on the topological influences of



**FIGURE 8.** Comparing the clustering methods and categories that maximize the dynamic similarity objective function: (a) average F1-score of the clustering methods, (b) average F1-score of the clustering categories, (c) average ARI of the clustering methods, (d) average ARI of the clustering categories, (e) average execution time (seconds) of the clustering methods, and (f) avg execution time of the clustering categories.

these edges, because they control the flow of information in the network.

As the results of the methods and categories that optimize the dynamic similarity objective function plotted in Fig. 8 show, the network centric clustering category (represented by the vertex influence heuristic method) outperformed the vertex centric clustering category (represented by the vertex random walk distance, vertex reachability, vertex degree, vertices complete mutuality, and vertex class membership methods). The dynamic similarity objective function seeks, mainly, to infer the global relative influences of vertices by analyzing their interaction patterns. The ultimate objective of the function is the identification of the influences of vertices to infer the social status of the individuals represented by these vertices: it assumes that if some vertices behave in the same role during interaction, they are likely to be similar. Based on our analysis of the experimental results,



**FIGURE 9.** Comparing the methods that maximizes the partition separability objective function, where: (a) F1-score of the clustering methods, (b) ARI of the clustering methods, (c) average execution time (in seconds) of the clustering methods.

we observed that the underlying techniques adopted by the vertex influence heuristic method achieved this objective by successfully inferring the influences of vertices. It did so by considering the topology of the entire network to analyze the interaction patterns of vertices. Specifically, the Kempe *et al.* [61] method considered the issue of selecting the set of influential individuals in a network as a discrete optimization problem. It assigned scores to vertices to reflect their global interaction roles and relative influences in networks. It successfully drew the boundaries of communities by taking into consideration the impact of their influential vertices.

From among the methods that maximize the partition separability objective function, the matrix-based eigenvectors method outperformed the normalized cost search, ratio cut search, and cut cost search methods as Fig. 9 shows. Based on our observation of the results, we attributed the performance

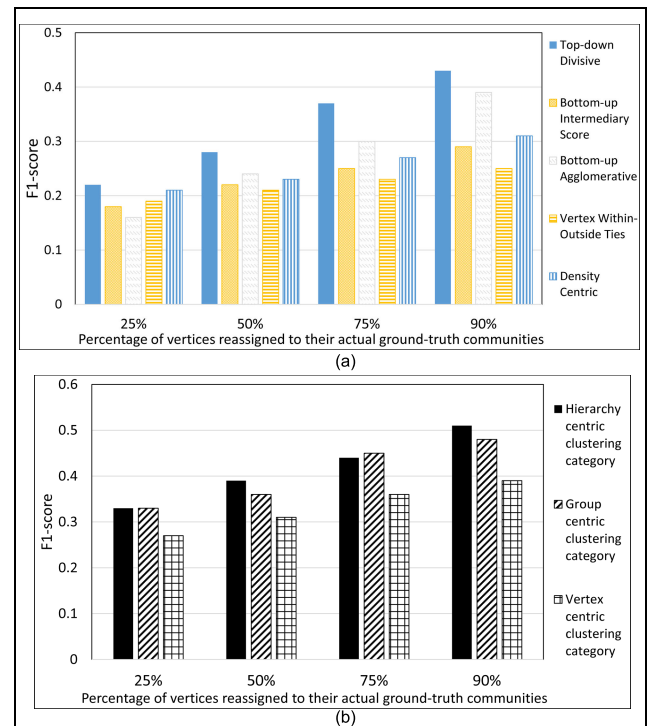
of the matrix-based eigenvectors method, mainly, to the fact that the Eigen spectrum of the modularity matrix employed by the method was closely tied to the detected communities' structures. We observed that the magnitudes of eigenvectors' elements reflected the strength of the "belonging" of vertices to the communities assigned to them. Eigenvectors contained important information about communities' structures. The matrix eigenvectors method maximized each group of vertices connected with edges that have higher than average density.

### B. EVALUATING THE ACCURACY OF ASSIGNING NEW VERTICES TO EXISTING COMMUNITIES

In real-world settings, there are always *new members* joining social networks that have *existing* communities. This makes it necessary for clustering methods to be able to advice such new members with the appropriate existing communities that match their profiles. Towards this, we evaluated the accuracy of the different methods and clustering categories to correctly assigning new vertices to the existing communities that match their structural profiles. We adopted the following strategy for the evaluation: (1) we shrank the boundaries of the com-LiveJournal communities (recall Table 1), (2) we considered the vertices that are outer of the new boundaries as new vertices need to be assigned to communities, and (3) we evaluated the accuracy of the methods for reassigning these vertices to their actual com-LiveJournal communities. We kept shrinking the boundary of each com-LiveJournal community until the percentage of its vertices that fell outside its new boundary amounted to 25% of its overall number of vertices. We considered the 25% as vertices that need to be reassigned to their actual ground-truth communities. We evaluated the accuracy of the methods to correctly reassign these vertices to their actual communities. We repeated the same procedure several times by reshinking the boundary of each com-LiveJournal community so that the percentage of its vertices fell outside its new boundary amounted to 50%, 75%, and then 90% of its overall number of vertices. Figs. 10-13 plot the results of comparing the methods and clustering categories in terms of their F1-scores for maximizing the internal density, structural similarity, dynamic similarity, and partition separability objective functions, respectively.

For maximizing the internal density objective function, the hierarchy centric, group centric, and vertex centric clustering categories achieved close results when the fraction of revealed vertices was small (see Fig. 10). However, as the fraction increases, the hierarchy centric and group centric categories kept outperforming the vertex centric clustering category at higher rate. Based on our analysis of the results, we attributed these findings to the following:

- Every time a new set of vertices was revealed, the hierarchy centric and group centric methods recomputed the relative influence of each vertex accordingly based on the topology of the entire network. These newly recomputed values were enhancements over the previous ones.

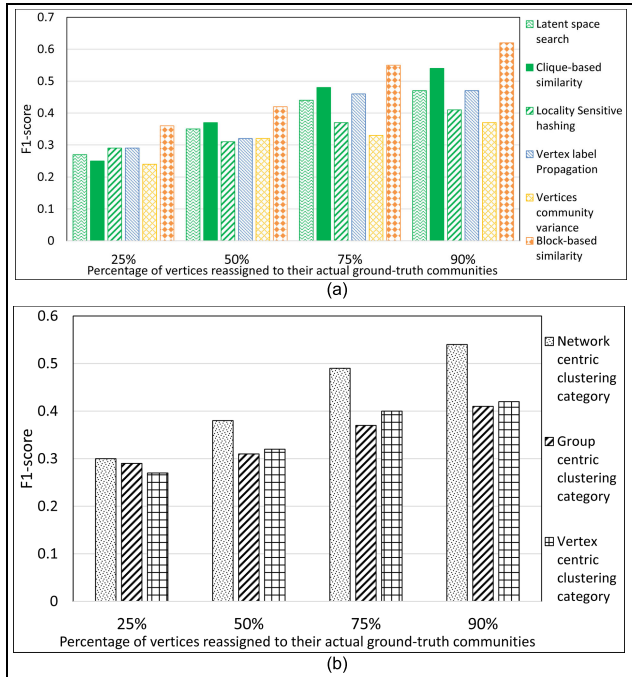


**FIGURE 10.** Comparing the clustering methods and categories that maximize the internal density objective function, where: (a) average F1-score of each clustering method to correctly reassign the vertices of com-LiveJournal dataset to their actual communities, and (b) average F1-score of each clustering category to correctly reassign the vertices of com-LiveJournal dataset to their actual communities.

In other words, every time a new set of vertices was revealed, the hierarchy centric and group methods optimized and updated the current relative influence scores of the vertices confined within a community. This is advantageous to the hierarchy centric and group centric methods, because their performances will keep improving over time, which is due to the fact that there are always new members joining social networks that have existing communities in a real-world setting.

- The group centric methods are better suited for detecting overlapping communities. Most of them are better suited for clustering together the sets of vertices whose neighborhoods are overlapping. These methods achieved good results, because many of the users of the com-LiveJournal dataset are members of more than one activity (i.e., community). LiveJournal classifies groups based on their activities into culture, sports, life/style, entertainment, gaming, technology, and student life.

For maximizing the structural similarity objective function, as the fraction of revealed vertices increases, the network centric category kept outperforming the vertex centric clustering category at higher rate as shown in Fig. 11. From our observation of the results, we attributed the insignificant accuracy increase rate of the vertex centric category to the following: the accuracy of the vertex centric category tended to degrade, if there is a large portion of a set of vertices that

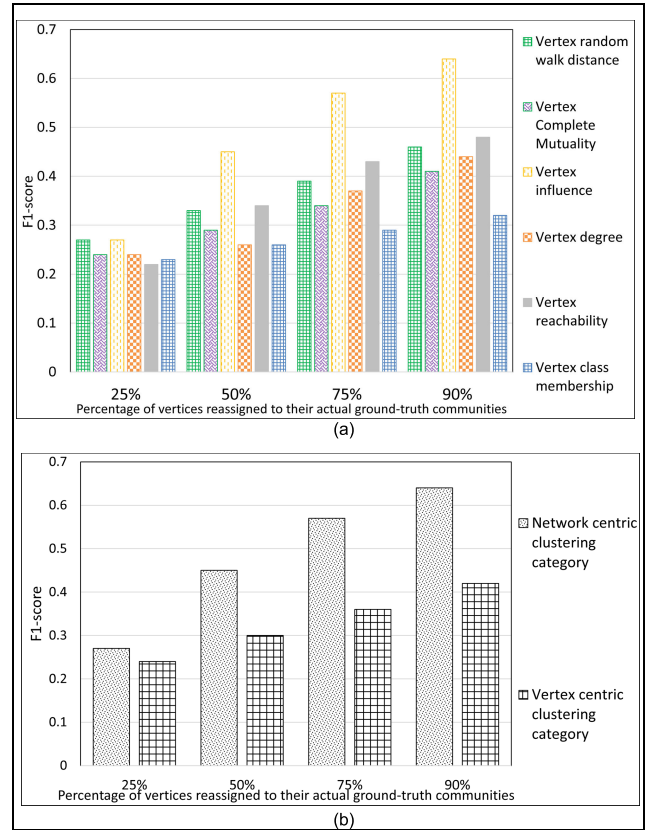


**FIGURE 11.** Comparing the clustering methods and categories that maximize the structural similarity objective function, where: (a) average F1-score of each clustering method to correctly reassign the vertices of com-LiveJournal dataset to their actual communities, and (b) average F1-score of each clustering category to correctly reassign the vertices of com-LiveJournal dataset to their actual communities.

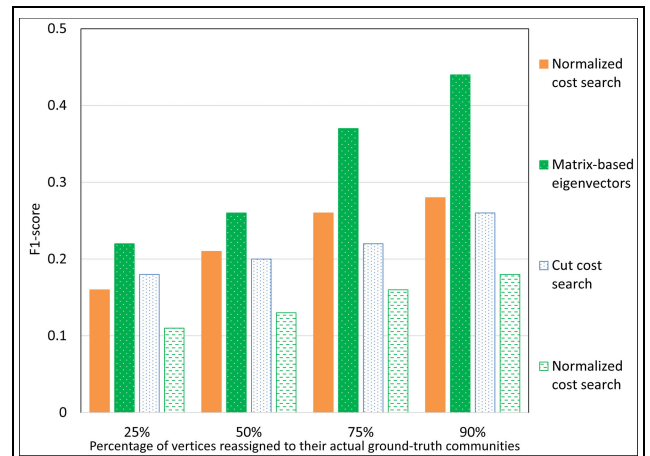
belongs to a community was also relevant to one or more other communities. This could be an indicative of a limitation of the category for detecting overlapping communities in large and complex networks.

For maximizing the dynamic similarity objective function, the network centric clustering category achieved good results as shown in Fig. 12. From our observation of the results, we attributed this performance to the fact that most of the methods falling under this category are better suited for detecting overlapping communities and a large number of the users of the com-LiveJournal dataset are members of more than one community (i.e., activity). Edges in most of these methods represent the intensities of the overlap between subgraphs (e.g., cliques and blocks) in a network. They identify a subgraph-subgraph overlap to detect overlapping communities.

From among the methods that maximize the partition separability objective function (see Fig. 13), we observed that the matrix eigenvectors method was able to accurately detect both group and individual vertices that are relevant to a specific community. It was able to select the vertices that maximize a community’s flow of information. As more information was revealed, the techniques of the method led to partitioning with higher correlations. We observed that the clustering accuracy of the normalized cost search method tended to degrade in situations where a set of revealed vertices was relevant to a specific community, but some of the vertices in this set were *more* relevant to other communities.



**FIGURE 12.** Comparing the clustering methods and categories that maximize the dynamic similarity objective function, where: (a) average F1-score of each clustering method to correctly reassign the vertices of com-LiveJournal dataset to their actual communities, and (b) average F1-score of each clustering category to correctly reassign the vertices of com-LiveJournal dataset to their actual communities.



**FIGURE 13.** Comparing the clustering methods that maximize the partition separability objective function in terms of the average F1-score of each method to correctly reassign the vertices of com-LiveJournal dataset to their actual communities.

### IX. EVALUATING THE DYNAMIC COMMUNITY DETECTION METHODS EMPIRICALLY AND EXPERIMENTALLY

We empirically and experimentally evaluated and compared the dynamic community detection methods that optimize the structural similarity and internal density objective functions.

## A. COMPILING GROUND TRUTH COMMUNITIES

### 1) SYNTHETIC DATASET

We employed the data generating method proposed by Newman and Girvan [87] for generating two types of synthetic datasets, one whose number of communities is fixed (called FIX-NUM-COM) and the other whose number of communities is variable (called VAR-NUM-COM). Below are descriptions of the two datasets.

- **FIX-NUM-COM:** The dataset comprises 128 nodes divided into four communities. Each community contains 32 nodes. We generated a network data for 10 consecutive timestamps. Dynamics into the generated data is introduced as follows. We selected three nodes at random from each of the communities in the 10 timestamps. From each community, we made three nodes leave the community and join the other three communities at random. Edges are placed randomly with a higher probability  $pin$  between two nodes within a same community and lower probability  $pout$  between-community nodes. The value of  $pout$  and  $pin$  are selected in such a way that the average node's degree is 16. To control the level of noise in the dynamic network, we considered a parameter  $z$  that reflects the average number of edges connecting between-community nodes. Specifically, we considered the following two values for  $z$ :

- $z = 5$ : for evaluating clear and easy to detect community structure due to low level of noise. This corresponds to  $pout = 0.16$  and  $pin = 0.05$ .
- $z = 6$ : for evaluating fuzzier community structure due to higher level of noise. This corresponds to  $pout = 0.12$  and  $pin = 0.06$ .

- **VAR-NUM-COM:** In this dataset, nodes are allowed to depart their original communities and form new ones. The dataset comprises 320 nodes divided into four communities. Each community contains 80 nodes. We generated a network data for 10 consecutive timestamps. We selected 12 nodes from each community at random and constructed two new communities from these nodes, each contained 24 nodes. This process was repeated in the first five timestamps. Then, the process was reversed in each of the other five timestamps, where nodes returned to their original communities. Therefore, the number of communities in timestamps 1-10 are 4, 6, 8, 10, 12, 12, 10, 8, 6, and 4 respectively. The values of  $pout$  and  $pin$  are selected in such a way that the average node's degree is half of the number of nodes in a community.

### 2) REAL DATASET

To further evaluate more features of the methods, we used the DBLP real-word dataset [118]. This dataset comprises 13,470 ground-truth communities constructed from the co-authorship of research papers in computer science. The dataset contains 317,080 nodes representing authors and 1,049,866 edges connecting the nodes. Two nodes in the co-authorship network are linked by an edge, if the two authors depicted by the two nodes published one or more papers together. Authors who published in a same

**TABLE 3.** The list of papers, in which the *selected* dynamic clustering techniques were proposed.

Objective function	Clustering category	Clustering method	Representative technique
Maximizing Internal Density	Using Auxiliary Update	Modularity Maximization	[101]
		Eigenvector Centrality	[50]
		Spectral Clustering	[28]
		Nonnegative Matrix Factorization	[49]
	Using Direct Update	Evolutionary Clustering	[41]
		Label Propagation	[140]
Maximizing Structural Similarity	Using Auxiliary Update	Block-based Clustering	[139]
		Clique-based Clustering	[31]
	Using Direct Update	Incremental-based Clustering	[75]
		Decomposition-based Clustering	[110]

journal/conference form a community. A venue of a publication is also a ground-truth community.

## B. EVALUATION SETUP

We performed the following procedure for the evaluations:

- 1) For each of the dynamic clustering methods described in Sections II and III, we selected the most influential technique employing the underlying principles of the method to serve as a representative of the method. We based the influence of a technique on factors such as the number of citations of the paper in which the technique was proposed, its recency, and its degree of state of the art.
  - Table 3 shows the list of papers, in which the *selected* dynamic clustering techniques were proposed.
- 2) For each clustering category, we ranked the different methods that fall under the category. The ranking was performed by averaging the fitness scores achieved by the techniques employed by the methods that fall under the clustering category.
- 3) For each clustering objective function, we ranked the different clustering categories that fall under the objective function.

## C. EVALUATING THE METHODS THAT MAXIMIZE THE STRUCTURAL SIMILARITY OBJECTIVE FUNCTION

### 1) EVALUATION USING THE ACCURACY OF DETECTED COMMUNITIES

In this test, we use the Normalized Mutual Information (NMI) metric described in Section VI for measuring the accuracies of the dynamic methods that maximize the structural similarity objective function. We measured the accuracies of the methods for clustering the FIX-NUM-COM and VAR-NUM-COM datasets described in Subsection IX.A.1 based on their ground truth communities and the memberships of these communities at each timestamp. The values of NMI range from 0 to 1. The higher the value the better accuracy.

- Fig. 14 shows the accuracies of detecting communities using the FIX-NUM-COM dataset.

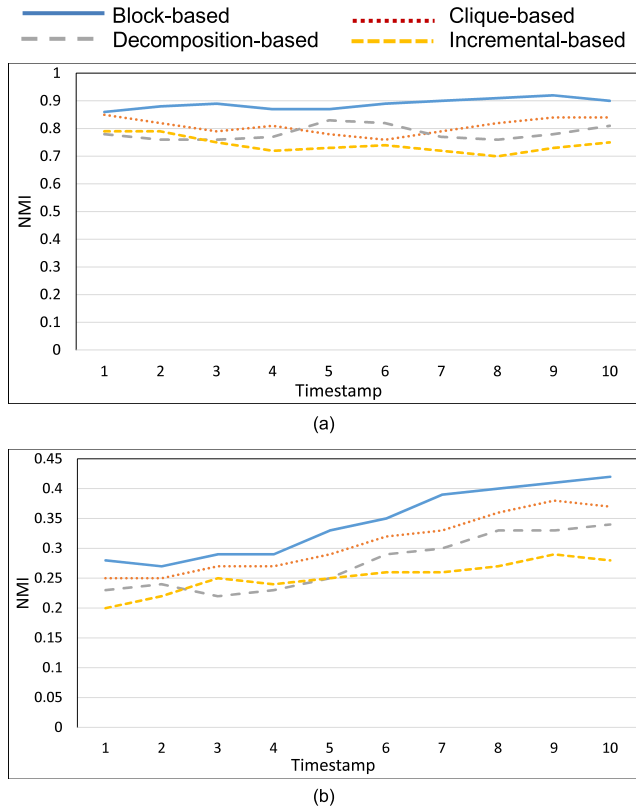


FIGURE 14. Accuracies of detecting communities using the FIX-NUM-COM dataset ( $\alpha = 0.8$ ), where: (a)  $z = 5$ , and (b)  $z = 6$ .

➤ Fig. 15 shows the accuracies of detecting communities using the VAR-NUM-COM dataset.

2) EVALUATING THE QUALITY OF SNAPSHOTS AND FREQUENCY OF FORMING AND DISSOLVING COMMUNITIES

Measuring the quality of snapshots reflects the degree in which detected clusters capture current network’s structure. Measuring the quality of temporal reflects the degree in which detected clusters are similar to previous clusters. During clustering, the trades of between the above two qualities at every timestamp results in smoothness. In this section, we evaluate the methods that maximize the structural similarity objective function using a parameter  $\alpha$ , which controls the following: (1) the trade-off between history quality and snapshot quality, and: (2) the frequency of dissolving/forming clusters. We used the DBLP dataset described in Subsection IX.A.B. Fig. 16 shows communities’ average length and the number of communities based on the variation of parameter  $\alpha$  from 0 to 1.

D. EVALUATING THE METHODS THAT MAXIMIZE THE INTERNAL DENSITY OBJECTIVE FUNCTION

In this test, we use Newman & G. modularity [87] and Chen *et al.* modularity [19] metrics described in Section VI for measuring the accuracies of the methods that maximize the internal density objective function. We used the

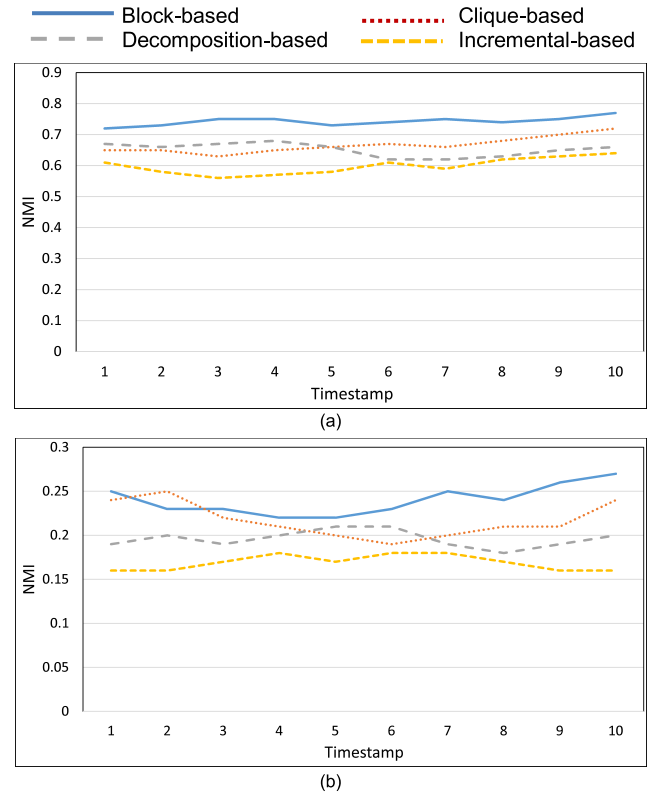


FIGURE 15. Accuracies of detecting communities in 10 timestamps using the VAR-NUM-COM dataset ( $\alpha = 0.8$ ), where: (a)  $z = 5$ , and (b)  $z = 6$ .

FIX-NUM-COM dataset described in Subsection IX.A.1 based on its ground truth communities and the memberships of these communities at each timestamp. Fig. 17 shows the modularity of detected communities in 10 timestamps, using Newman & G. modularity and Chen *et al.* modularity.

E. DISCUSSION OF THE RESULTS

Table 4 shows the fitness score of each technique representing a method, the ranking of the different methods that fall under a same clustering category, and the ranking of the different clustering categories that fall under a same objective function. In the next two subsections 1-4, we discuss our observation of the results of the methods and clustering categories that maximize the structural similarity and internal density objective functions.

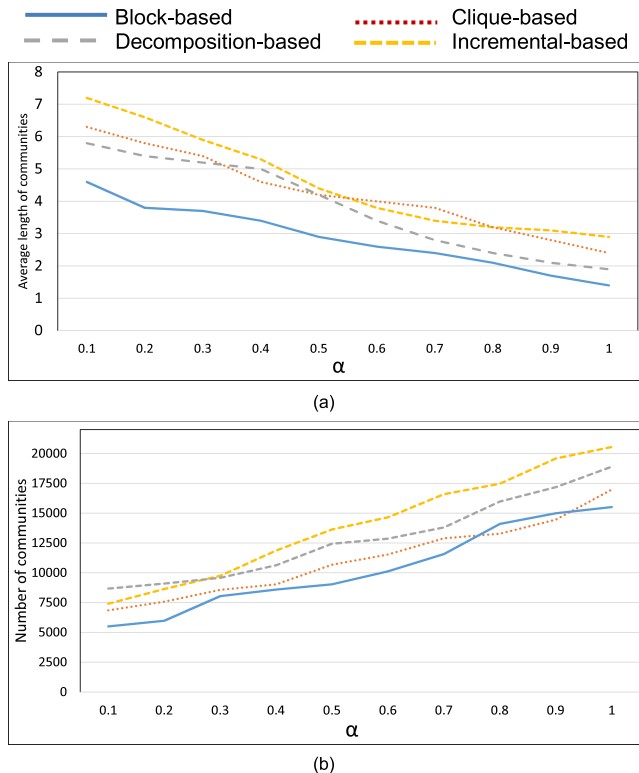
1) RESULTS OF THE METHODS THAT MAXIMIZE THE STRUCTURAL SIMILARITY

• *Using FIX-NUM-COM dataset:* When it is clear and easy to detect community structure due to low level of noise (i.e.,  $z = 5$ ), the results revealed that the block-based method achieved very good accuracy results. When community structures are fuzzier due to higher level of noise (i.e.,  $z = 6$ ), the results revealed that both of the block-based and clique-based methods achieved very good accuracy results. The good performance of the two methods is attributed, mainly, to the temporal smoothing techniques employed by

**TABLE 4.** The fitness score of each technique representing a dynamic method, the ranking of the different methods that fall under a same clustering category, and the ranking of the different clustering categories that fall under a same objective function.

Objective function	Clustering category	Clustering method	Rep. tech.	Fitness metrics for the objective functions	Avg. Tech. score	Method rank	Category rank
Maximizing Internal Density	Using Auxiliary Update	Modularity Maximization	[101]	Newman & G. modularity	0.540	2	2
			[50]	Chen et al. modularity	0.408		
		Eigenvector Centrality	[50]	Newman & G. modularity	0.632	1	
			[50]	Chen et al. modularity	0.571		
		Spectral Clustering	[28]	Newman & G. modularity	0.505	3	
			[28]	Chen et al. modularity	0.396		
	Nonnegative Matrix Factorization	[49]	Newman & G. modularity	0.449	4		
			Chen et al. modularity	0.371			
	Using Direct Update	Evolutionary Clustering	[41]	Newman & G. modularity	0.664	1	1
			[41]	Chen et al. modularity	0.622		
Label Propagation	[140]	Newman & G. modularity	0.551	2			
		Chen et al. modularity	0.475				
Maximizing Structural Similarity	Using Auxiliary Update	Block-based Clustering	[139]	Normalized mutual information	0.550	1	1
			[139]	Smoothness change (com. avg length)	0.356		
		Clique-based Clustering	[31]	Normalized mutual information	0.501	2	
			[31]	Smoothness change (com. avg length)	0.431		
	Using Direct Update	Incremental-based Clustering	[75]	Normalized mutual information	0.441	2	2
			[75]	Smoothness change (com. avg length)	0.478		
		Decomposition-based Clustering	[110]	Normalized mutual information	0.477	1	
				Smoothness change (com. avg length)	0.433		

"Rep.", avg, and "tech." are abbreviations of the terms "representative", average, and "technique" respectively



**FIGURE 16.** (a) Communities' average length, and (b) number of communities. (a) and (b) are based on parameter  $\alpha$  variation from 0 to 1.

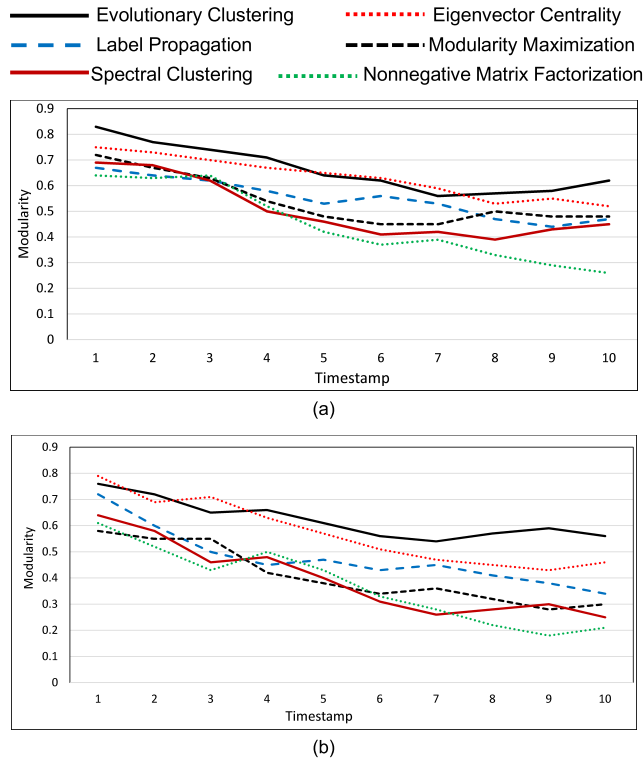
the methods. This is evident in the accuracy increase at each timestamp  $t_s$  over timestamp  $t_s-1$  due to the improvement of smoothness.

- *Using VAR-NUM-COM dataset:* The block-based method showed significant performance over the other methods in the two values of parameter  $z$  (i.e.,  $z = 5$  and  $z = 6$ ). This is because the method is sensitive to any change occurs in the number of communities due to the dissolving of communities and the formation of new communities. The method deals with such changes effectively. However, the performance of the method at the middle of the timestamps was not as good as the beginning and end when  $z = 6$ . This is because the difference between the number communities produced by the method and the ground truth data is maximized at the middle. The incremental-based method did not respond effectively to changes in number of communities. The decomposition method showed performance degradation between timestamps 7 and 10. The accuracy of all methods when  $z = 6$  was much lower than when  $z = 5$ . This due to the lack of temporal smoothness as a result of constant changes in the number of communities. As Fig. 16 shows, the block-based method showed the best stability in terms of number of detected communities and the lengths of these communities as timestamp goes on. This is an indicative of smoothness in the detection of communities.

## 2) RESULTS OF THE METHODS THAT MAXIMIZE THE INTERNAL DENSITY

As Fig. 17 shows, all methods achieved good modularity values at the beginning of the timestamps. This is because local clusters are likely to be connected and the lifetimes of communities tend to be high (i.e., the variation of number of communities tends to be low) at the beginning of the





**FIGURE 17.** The modularity of detected communities in 10 timestamps, using: (a) Newman & G. modularity, and (b) Chen *et al.* modularity.

timestamps. However, as timestamp goes on, the modularity of all the methods decreases. This is because the number of local clusters that are disconnected increases as timestamp goes on due to their low densities. Therefore, the number of communities increases and a community's lifetime decreases as timestamp goes on.

## X. CONCLUSION

### A. SUMMARY

Most current survey papers classify community detection methods into broad categories and do not draw clear boundaries between the specific techniques employed by these methods. To overcome this, we introduced this survey paper to classify static and dynamic clustering techniques into fine-grained categories and methods. We provided methodology-based taxonomies that classify static and dynamic community detection methods into hierarchically nested, fine-grained, and specific classes. We provided taxonomies, whose classifications resulted in 31 fine-grained methods for detecting clusters. We hierarchically classified the clustering methods and categories that optimize each objective function. We empirically and experimentally compared and ranked the different methods that fall under each clustering category. We also empirically and experimentally compared and ranked the different clustering categories that optimize a same objective function. We summarize below the findings of the empirical and experimental evaluations:

- **Findings of the static and dynamic methods that maximize the structural similarity objective function:**
  - *Static community detection methods:*
    - The network centric clustering category achieved better results than the vertex centric and group centric clustering categories.
    - Among the clustering methods that fall under the network centric clustering category, the block-based sub-graph similarity method achieved better results than the clique-based sub-graph similarity and latent space search methods.
    - The vertex centric clustering category has a limitation in detecting overlapping communities for large and complex networks.
  - *Dynamic community detection methods:*
    - The block-based method achieved very good accuracy results when it is clear and easy to detect community structure.
    - The block-based and clique-based methods achieved very good accuracy results when community structures are fuzzier.
    - The block-based method showed significant performance over the other methods using a dataset, whose number of communities is variable.
    - The incremental-based method did not respond effectively to changes in number of communities.
- **Findings of the static and dynamic methods that maximize the internal density objective function:**
  - *Static community detection methods:*
    - The hierarchy centric and group centric clustering categories outperformed the vertex centric category. Results showed that the performance of the hierarchy centric and group centric categories kept improving as the sizes of communities increased. This is advantageous to the two methods because communities' sizes keep increasing over time in real word setting.
    - Among the clustering methods that fall under the hierarchy centric clustering category, the top-down divisive-based clustering method achieved better results that the bottom-up agglomerative and the bottom-up intermediary score maximization clustering methods.
  - *Dynamic community detection methods:*
    - All methods achieved good modularity values at the beginning of the timestamps. However, as timestamp goes on, the modularity of all the methods decreases.
    - The number of communities increases and a community's lifetime decreases as timestamp goes on.

- **Findings of the static methods that maximize the dynamic similarity objective function:**

- The network centric category achieved better results than the vertex centric clustering category. Results showed that the network centric category kept outperforming the vertex centric category at higher rate as communities' sizes increased.
- The network centric methods are better suited for detecting overlapping communities in large and complex networks.
- Among the clustering methods that fall under the vertex centric clustering category, the vertex random walk distance method achieved better results than the vertex reachability, vertex degree, vertex complete mutuality, and vertex class membership methods.

- **Findings of the static methods that maximize the partition separability objective function:**

- Among the methods that fall under the network centric category, the matrix-based eigenvectors method outperformed the normalized cost search, ratio cut search, and cut cost search methods.
- The normalized cost search method was able to accurately detect both, group and individual vertices that were relevant to a specific community. However, the accuracy of the method tended to degrade in situations where a set of vertices was relevant to a specific community, but some of them were *more* relevant to other communities.

- Maximizing the partition separability objective function can be achieved by adopting the underlying techniques of the network-centric clustering category (recall Subsections II.A.1 and V.B). These techniques ensure that each pair of connected vertices in a partition is closely associated. A global preprocessing scheme should be employed for identifying closely associated vertices based on the topology of the entire network.
- Maximizing the dynamic similarity objective function can be achieved by adopting the underlying techniques of the hierarchy-centric clustering category (recall Subsection III.A.1). One of these techniques employs a ranking scheme that assigns a score to each vertex to reflect its global relative importance and degree of interaction role in the network.
- Maximizing the internal density objective function can be achieved by adopting the underlying techniques of the group-centric clustering category (recall Subsections 2.1.3 and 3.1.3). One of these techniques takes into consideration the associations between all connections confined within a partition. This ensures that each pair of vertices within a partition is closely associated.
- Maximizing the structural similarity objective function can be achieved by adopting the underlying techniques of the vertex-centric clustering category (recall Subsection 3.1.2). One of these techniques considers a pair of vertices to be part of a partition, only if the degrees of association between the pair and the influential vertices in the partition are significant.

## B. CURRENT CHALLENGES AND POSSIBLE SOLUTIONS

The results of the empirical and experimental evaluations revealed the following two major limitations, which negatively affected the quality of a large number of communities detected by different methods: (1) inconsideration of multi-objective functions, and (2) inconsideration of how closely associated vertices are based on the global influences of the edges connecting them. We discuss below these two limitations and present our recommended solutions.

### 1) INCONSIDERATION OF MULTI-OBJECTIVE FUNCTIONS

Real-world networks are complex and may require *multiple* driving factors of quality partitioning. One of these factors is the optimization of *multiple* objective functions. Realizing a “good” community by solely optimizing a single objective function is often an unrealistic expectation [72]. This may cause a method to work well for *only* certain settings of real-world networks. To overcome this, many methods adopted multi-objective techniques (e.g., [133]). However, most of these methods optimize only two functions. We observed that most real-world networks require maximizing two or more of the following objective functions: internal density, structural similarity, dynamic similarity, and partition separability.

Below are our recommended solutions for maximizing each of the four objective functions:

### 2) INCONSIDERATION OF HOW CLOSELY ASSOCIATED VERTICES ARE BASED ON THE GLOBAL INFLUENCES OF THE EDGES CONNECTING THEM

By analyzing the results of the empirical and experimental evaluations, we observed that most of the methods detected communities in the independence of how closely related their connections are based on the global relative importance of the edges connecting them. They considered all edges to have the same degree of influence. Intuitively, however, some communication channels (i.e., edges) pass significant amount of the information diffused by the influential individuals (i.e., influential vertices) in a social network, while others do not. Consequently, most of these methods may not work well in networks with connections that have varying degrees of association with the influential vertices in the networks. To overcome this, a good method should be able to take into consideration the impact of the global influences of edges on the degrees of association between the vertices at the endpoints of these edges.

## REFERENCES

- [1] N. Agarwal, H. Liu, L. Tang, and P. S. Yu, “Identifying the influential Bloggers in a community,” in *Proc. Int. Conf. Web Search Web Data Mining*, New York, NY, USA, 2008, pp. 207–218.
- [2] R. Andersen, F. Chung, and K. Lang, “Local graph partitioning using PageRank vectors,” in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci. (FOCS)*, Oct. 2006, pp. 475–486.

- [3] A. Airodi, D. Blei, S. Fienberg, and E. Xing, "Mixed membership stochastic block models," *J. Mach. Learn. Res.*, vol. 9, pp. 1981–2014, 2008.
- [4] M. Alzaabi, K. Taha, and T. A. Martin, "CISRI: A crime investigation system using the relative importance of information spreaders in networks depicting criminals communications," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2196–2211, Oct. 2015.
- [5] A. Angel, N. Sarkas, N. Koudas, and D. Srivastava, "Dense subgraph maintenance under streaming edge weight updates for real-time story identification," *Proc. VLDB Endowment*, vol. 5, no. 6, pp. 574–585, Feb. 2012.
- [6] T. Aynaud and J.-L. Guillaume, "Static community detection algorithms for evolving networks," in *Proc. 8th IEEE Int. Symp. Modeling Optim. Mobile, Ad Hoc Wireless Netw. (WiOpt)*, May 2010, pp. 513–519.
- [7] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 4, pp. 16:1–16:36, Nov. 2009.
- [8] R. Agrawal, "Bi-objective community detection (BOCD) in networks using genetic algorithm," in *Contemporary Computing*. Berlin, Germany: Springer-Verlag, 2011, pp. 5–15.
- [9] B. Balasundaram, S. Butenko, and I. V. Hicks, "Clique relaxations in social network analysis: The maximum-plex problem," *Oper. Res.*, vol. 59, no. 1, pp. 133–142, Feb. 2011.
- [10] S. Borgatti, M. Everett, and P. Shirey, "LS sets, lambda sets and other cohesive subsets," *Social Netw.*, vol. 12, no. 4, pp. 337–357, 1990.
- [11] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 92–100.
- [12] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.
- [13] S. Bhatt, S. Padhee, A. Sheth, K. Chen, V. Shalin, D. Doran, and B. Minnery, "Knowledge graph enhanced community detection and characterization," in *Proc. 12th ACM Int. Conf. Web Search Data Mining (WSDM)*, Melbourne, Australia, Jan. 2019.
- [14] A. Bahulkar, B. K. Szymanski, N. O. Baycik, and T. C. Sharkey, "Community detection with edge augmentation in criminal networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Barcelona, Spain, Aug. 2018, pp. 1168–1175.
- [15] P. Bogdanov, M. Mongiovi, and A. K. Singh, "Mining heavy subgraphs in time-evolving networks," in *Proc. IEEE 11th Int. Conf. Data Mining*, Dec. 2011, pp. 81–90.
- [16] J. Chen and Y. Saad, "Dense subgraph extraction with application to community detection," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 7, pp. 1216–1230, Jul. 2012.
- [17] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, pp. 98–101, May 2008.
- [18] M. Coscia, F. Giannotti, and D. A. Pedreschi, "Classification for community discovery methods in complex networks," *CoRR*, vol. abs/1206.3552, 2012.
- [19] D. Chen, M. Shang, Z. Lv, and Y. Fu, "Detecting overlapping communities of weighted networks via a local algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 389, no. 19, pp. 4177–4187, Oct. 2010.
- [20] M. Crampes and M. Plantié, "A unified community detection, visualization and analysis method," *CoRR*, vol. abs/1301.7006, 2013.
- [21] M. Chen, T. Nguyen, and B. Szymanski, "A new metric for quality of network community structure," *ASE Hum. J.*, vol. 2, no. 4, pp. 226–240, 2013.
- [22] G. Caldarelli, R. Pastor-Satorras, and A. Vespignani, "Cycles structure and local ordering in complex networks," 2002, *arXiv:cond-mat/0212026*. [Online]. Available: <https://arxiv.org/abs/cond-mat/0212026>
- [23] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, Dec. 2004, Art. no. 066111.
- [24] W.-K. Cui, K.-K. Shang, Y.-J. Zhang, J. Xiao, and X.-K. Xu, "Constructing null networks for community detection in complex networks," *Eur. Phys. J. B*, vol. 91, no. 7, p. 145, Jul. 2018.
- [25] R. Cazabet, F. Amblard, and C. Hanachi, "Detection of overlapping communities in dynamical social networks," in *Proc. IEEE 2nd Int. Conf. Social Comput. (SocialCom)*, vol. 14, Aug. 2010, p. 309.
- [26] G. Murray, G. Carenini, and R. Ng, "Using the omega index for evaluating abstractive community detection," in *Proc. Workshop Eval. Metrics Syst. Comparison Autom. Summarization, Assoc. Comput. Linguistics*, 2012, pp. 10–18.
- [27] J. Hou Chin and K. Ratnavelu, "A semi-synchronous label propagation algorithm with constraints for community detection in complex networks," *Sci. Rep.*, vol. 7, no. 1, Apr. 2017, Art. no. 45836.
- [28] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng, "Evolutionary spectral clustering by incorporating temporal smoothness," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 153–162.
- [29] D. Chakrabarti, R. Kumar, and A. S. Tomkins, "Evolutionary clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, pp. 554–560. ACM Press, 2006.
- [30] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Mixing local and global information for community detection in large networks," *J. Comput. Syst. Sci.*, vol. 80, no. 1, pp. 72–87, Feb. 2014.
- [31] D. Duan, Y. Li, R. Li, and Z. Lu, "Incremental K-clique clustering in dynamic social networks," *Artif. Intell. Rev.*, vol. 38, no. 2, pp. 129–147, Aug. 2012.
- [32] T. N. Dinh, Y. Xuan, and M. T. Thai, "Towards social-aware routing in dynamic communication networks," in *Proc. 28th Int. Perform. Comput. Commun. Conf.*, Dec. 2009, pp. 161–168.
- [33] M. G. Everett and S. P. Borgatti, "Regular equivalence: General theory," *J. Math. Sociol.*, vol. 19, no. 1, pp. 29–52, May 1994.
- [34] P. Erdős and A. Rényi, *On Random Graphs*, vol. 6. Debrecen, Hungary: Publicationes Mathematicae, 1959, pp. 290–297.
- [35] G. Flake, S. Lawrence, and C. Giles, "Efficient identification of Web communities," in *Proc. 6th ACM SIGKDD*, New York, NY, USA, 2000, pp. 150–160.
- [36] K. Faust and S. Wasserman, "Blockmodels: Interpretation and evaluation," *Social Netw.*, vol. 14, nos. 1–2, pp. 5–61, Mar. 1992.
- [37] I. Farkas, D. Ábel, G. Palla, and T. Vicsek, "Weighted network modules," *New J. Phys.*, vol. 9, no. 6, p. 180, Jun. 2007.
- [38] A. Fang-ju, "Research on a large-scale community detection algorithm based on non-weighted graph," *Cluster Comput.*, vol. 22, no. S2, pp. 2555–2562, Mar. 2019.
- [39] T. Falkowski, A. Barth, and M. Spiliopoulou, "Dengraph: A density-based community detection algorithm," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, Nov. 2007, pp. 112–115.
- [40] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou, "Mining and visualizing the evolution of subgroups in social networks," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, Dec. 2006, pp. 52–58.
- [41] F. Folino and C. Pizzuti, "An evolutionary multiobjective approach for community discovery in dynamic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1838–1852, Aug. 2014.
- [42] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 1, pp. 36–41, 2007.
- [43] M. Gong, Q. Cai, Y. Li, and J. Ma, "An improved memetic algorithm for community detection in complex networks," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Brisbane, QLD, Australia, Jun. 2012, pp. 1–8.
- [44] S. Giannini, "RDF data clustering," in *Proc. BI Syst. Workshops*, Poznan, Poland, 2013.
- [45] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Jun. 2002.
- [46] C. Gong, G. Wang, J. Hu, M. Liu, L. Liu, and Z. Yang, "Finding multi-granularity community structures in social networks based on significance of community partition," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Singapore, Nov. 2018, pp. 415–421.
- [47] R. Görke, T. Hartmann, and D. Wagner, "Dynamic graph clustering using minimum-cut trees," *J. Graph Algorithms Appl.*, vol. 16, no. 2, pp. 411–446, 2012.
- [48] R. Görke, P. Maillard, A. Schumm, C. Staudt, and D. Wagner, "Dynamic graph clustering combining modularity and smoothness," *J. Exp. Algorithmics*, vol. 18, no. 1, pp. 1:1–1:29, Dec. 2013.
- [49] L. Gauvin, A. Panisson, and C. Cattuto, "Detecting the community structure and activity patterns of temporal networks: A non-negative tensor factorization approach," *PLoS ONE*, vol. 9, no. 1, Jan. 2014, Art. no. e86028.
- [50] P. Guan and J. Wu, "Effective data communication based on social community in social opportunistic networks," *IEEE Access*, vol. 7, pp. 12405–12414, 2019.
- [51] R. Görke, P. Maillard, C. Staudt, and D. Wagner, "Modularity-driven clustering of dynamic graphs," in *Proc. SEA*, vol. 6049, 2010, pp. 436–448.
- [52] M. Handcock, A. Raftery, and J. Tantrum, "Model-based clustering for social networks," *J. Roy. Stat. Soc. A*, vol. 127, no. 2, pp. 301–354, 2007.

- [53] R. A. Hanneman and M. Riddle, "Introduction to social network methods," Univ. California, Riverside, Riverside, CA, USA, Tech. Rep., 2005.
- [54] S. Harenberg, G. Bello, L. Gjeltema, J. Harlalka, R. Seay, and N. Samatova, *Community Detection in Large-Scale Networks: A Survey and Empirical Evaluation*. Hoboken, NJ, USA: Wiley, 2014.
- [55] J. E. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," *Proc. Nat. Acad. Sci. USA*, vol. 101, pp. 5244–5253, Apr. 2004.
- [56] A. Jain and R. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.
- [57] P. Jiang and M. Singh, "SPiCi: A fast clustering algorithm for large biological networks," *Bioinformatics*, vol. 26, no. 8, pp. 1105–1111, Apr. 2010.
- [58] Y. Jaewon and L. Jure, "Defining and evaluating network communities based on ground-truth," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2012, pp. 745–754.
- [59] M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "CatchSync: Catching synchronized behavior in large directed graphs," in *Proc. 20th ACM Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2014, pp. 941–950.
- [60] L. Jiang, L. Shi, L. Liu, J. Yao, and M. Yousef, "User interest community detection on social media using collaborative filtering," *Wireless Netw.*, vol. 25, no. 7, p. 4443, 2019.
- [61] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 137–146.
- [62] B. Kernighan and S. Lin, "An efficient heuristic for partitioning graphs," *Bell Syst. Tech. J.*, vol. 49, pp. 291–308, 1970.
- [63] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Proc. 12th ACM SIGKDD*, New York, NY, USA, 2006, pp. 337–357.
- [64] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 359–392, Jan. 1998.
- [65] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki, "Sequential algorithm for fast clique percolation," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 2, Aug. 2008, Art. no. 026109.
- [66] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 1, Jan. 2011, Art. no. 016107.
- [67] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Trawling the Web for emerging cyber-communities," *Comput. Netw.*, vol. 31, nos. 11–16, pp. 1481–1493, May 1999.
- [68] A. Konstantinidis, D. Zeinalipour-Yazdi, P. Andreou, and G. Samaras, "Multi-objective query optimization in smartphone social networks," in *Proc. 12th Int. Conf. Mobile Data Manage. (MDM)*, 2011, pp. 27–32.
- [69] M. S. Kim and J. Han, "A particle-and-density based evolutionary clustering method for dynamic networks," in *Proc. 35th Int. Conf. Very Large Databases (VLDB)*, 2009, pp. 622–633.
- [70] R. Kannan, S. Vempala, and A. Vetta, "On clusterings: Good, bad and spectral," *J. ACM*, vol. 51, no. 3, pp. 497–515, 2004.
- [71] D. Liu, D. Jin, C. Baquero, D. He, B. Yang, and Q. Yu, "Genetic algorithm with a local search strategy for discovering communities in complex networks," *Int. J. Comput. Intell. Syst.*, vol. 6, no. 2, pp. 354–369, Apr. 2013.
- [72] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 695–704.
- [73] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBrienen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 420–429.
- [74] P.-Z. Li, L. Huang, C.-D. Wang, J.-H. Lai, and D. Huang, "Community detection by motif-aware label propagation," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 2, pp. 1–19, Mar. 2020.
- [75] W. Liu, T. Suzumura, L. Chen, and G. Hu, "A generalized incremental bottom-up community detection framework for highly dynamic graphs," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Boston, MA, USA, Dec. 2017, pp. 3342–3351.
- [76] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *ACM Trans. Knowl. Discovery Data*, vol. 3, no. 2, pp. 1–31, Apr. 2009.
- [77] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, Mar. 2009, Art. no. 033015.
- [78] K. Macropoul and A. Singh, "Scalable discovery of best clusters on large graphs," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 693–702, Sep. 2010.
- [79] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *CoRR*, vol. abs/1308.0971, 2013.
- [80] X. Meng, L. Dong, Y. Li, and W. W. Guo, "A genetic algorithm using K-path initialization for community detection in complex networks," *Cluster Comput.*, vol. 20, no. 1, pp. 311–320, Mar. 2017.
- [81] N. Mehrabi, F. Morstatter, N. Peng, and A. Galstyan, "Debiasing community detection: The importance of lowly connected nodes," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Vancouver, BC, Canada, 2019, pp. 509–512.
- [82] N. Pósfai, N. Braun, B. A. Beisner, B. McCowan, and R. M. D'Souza, "Consensus ranking for multi-objective interventions in multiplex networks," *New J. Phys.*, vol. 21, no. 5, May 2019, Art. no. 055001.
- [83] R. Márquez, "Overlapping community detection in static and dynamic networks," in *Proc. 13th Int. Conf. Web Search Data Mining (WSDM)*, Houston, TX, USA, Jan. 2020, pp. 925–926.
- [84] M. E. J. Newman and J. Park, "Why social networks are different from other types of networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 3, Sep. 2003, Art. no. 036122.
- [85] K. Nowicki and T. Snijders, "Estimation and prediction for stochastic block structures," *J. Amer. Stat. Assoc.*, vol. 96, no. 455, pp. 1077–1087, 2001.
- [86] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, Sep. 2006, Art. no. 036104.
- [87] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.
- [88] L. Ni, W. Luo, W. Zhu, and B. Hua, "Local overlapping community detection," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 1, pp. 1–25, Feb. 2020.
- [89] C.-C. Ni, Y.-Y. Lin, F. Luo, and J. Gao, "Community detection on networks with Ricci flow," *Sci. Rep.*, vol. 9, no. 1, pp. 1–2, Dec. 2019.
- [90] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. Huang, "Incremental spectral clustering with application to monitoring of evolving blog communities," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2007, pp. 261–272.
- [91] G. K. Orman, V. Labatut, and H. Cherifi, "Comparative evaluation of community detection algorithms: A topological approach," *J. Stat. Mech. Theory Exp.*, vol. 2012, no. 8, Aug. 2012, Art. no. P08001.
- [92] M. Porter, J. P. Onnela, and P. J. Mucha, "Communities in networks," *Notices Amer. Math. Soc.*, vol. 56, no. 9, pp. 1082–1097, 2009.
- [93] P. Pons, "Détection de communautés dans les grands graphes de terrain," Ph.D. dissertation, Paris Univ., Paris, France, 2007.
- [94] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," in *Proc. Data Mining Knowl. Discovery*, Apr. 2011, pp. 1–40.
- [95] G. Palla, A.-L. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, Apr. 2007.
- [96] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," in *Proc. Int. Conf. Parallel Problem Solving Nature*, Berlin, Germany, 2008.
- [97] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.
- [98] P. Pascal and M. Latapy, "Computing communities in large networks using random walks," in *Computer and Information Sciences—ISCIS*. Berlin, Germany: Springer, 2005, pp. 284–293.
- [99] T. P. Peixoto, "Merge-split Markov chain Monte Carlo for community detection," 2020, *arXiv:2003.07070*. [Online]. Available: <http://arxiv.org/abs/2003.07070>
- [100] J. Palowitch, S. Bhamidi, and A. Nobel, "Significance-based community detection in weighted networks," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 188:1–188:48, 2018.
- [101] C. Pizzuti and A. Socievole, "Many-objective optimization for community detection in multi-layer networks," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, San Sebastian, Spain, Jun. 2017, pp. 411–418.
- [102] S. Pang, C. Chen, and T. Wei, "A realtime community detection algorithm: Incremental label propagation," in *Proc. 1st Int. Conf. Future Inf. Netw.*, Oct. 2009, pp. 313–317.
- [103] X. Qian, L. Yang, and J. Fang, "Overlapping community detection based on community connection similarity of maximum clique," in *Proc. ICPC-SEE*, vol. 1, 2018, pp. 241–252.

- [104] X. Qian, L. Yang, and J. Fang, "Heterogeneous network community detection algorithm based on maximum bipartite clique," in *Proc. ICPC-SEE*, 2018, pp. 253–268.
- [105] J. Riedy, D. A. Bader, and H. Meyerhenke, "Scalable multi-threaded community detection in social networks," in *Proc. 26th IPDPSW*, Washington, DC, USA, May 2012, pp. 1619–1628.
- [106] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 1, 2006, Art. no. 016110.
- [107] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.
- [108] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, Sep. 2007, Art. no. 036106.
- [109] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *Eur. Phys. J. Special Topics*, vol. 178, no. 1, pp. 13–23, Nov. 2009.
- [110] G. R. Rossetti, "Graph benchmark handling community dynamics," *J. Complex Netw.*, vol. 5, no. 6, pp. 893–912, 2017.
- [111] R. Shang, J. Bai, L. Jiao, and C. Jin, "Community detection based on modularity and an improved genetic algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 392, no. 5, pp. 1215–1231, Mar. 2013.
- [112] V. Sindhvani and P. Niyogi, "A co-regularization approach to semisupervised learning with multiple views," in *Proc. ICML Workshop Learn. Multiple Views*, 2005, pp. 824–831.
- [113] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, p. 103, Jun. 2009.
- [114] P. Sarkar and A. Moore, "Dynamic social network analysis using latent space models," *SIGKDD Explor. Newsl.*, vol. 7, no. 2, pp. 31–40, 2005.
- [115] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–904, Aug. 2000.
- [116] A. Sankararaman and F. Baccelli, "Community detection on Euclidean random graphs," in *Proc. 29th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, New Orleans, LA, USA, 2018, pp. 2181–2200.
- [117] J. Shang, L. Liu, F. Xie, Z. Chen, J. Miao, X. Fang, and C. Wu, "A real-time detecting algorithm for tracking community structure of dynamic networks," *CoRR*, vol. abs/1407.2683, 2014.
- [118] *Stanford Large Network Dataset Collection*. Accessed: Mar. 16, 2020. [Online]. Available: <http://snap.stanford.edu/data/>
- [119] K. Taha, "Disjoint community detection in networks based on the relative association of members," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 2, pp. 493–507, Jun. 2018.
- [120] K. Taha and P. D. Yoo, "Shortlisting the influential members of criminal organizations and identifying their important communication channels," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 1988–1999, Aug. 2019.
- [121] K. Taha and P. D. Yoo, "Using the spanning tree of a criminal network for identifying its leaders," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 2, pp. 445–453, Feb. 2017.
- [122] K. Taha and R. Elmasri, "GOcSim: GO context-driven similarity," in *Proc. IEEE Symp. Comput. Intell. Bioinf. Comput. Biol. (CIBCB)*, San Diego, CA, USA, May 2012, pp. 355–362.
- [123] M. Tasgin, A. Herdagdelen, and H. Bingol, "Communities detection in complex networks using genetic algorithms," in *Proc. Eur. Conf. Complex Syst. (ECSS)*, 2006.
- [124] K. Taha, D. Homouz, H. Al Muhairi, and Z. Al Mahmoud, "GRank: A middleware search engine for ranking genes by relevance to given genes," *BMC Bioinf.*, vol. 14, no. 1, p. 251, Dec. 2013.
- [125] K. Taha, "Determining semantically related significant genes," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 6, pp. 1119–1130, Nov. 2014.
- [126] K. Taha and P. D. Yoo, "SIIMCO: A forensic investigation tool for identifying the influential members of a criminal organization," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 811–822, Apr. 2016.
- [127] K. Taha and R. Elmasri, "BusSEngine: A Business Search Engine, Knowledge and Information Systems," *Int. J.*, vol. 23, no. 2, pp. 153–197, 2010.
- [128] M. Takaffoli, J. Fagnan, F. Sangi, and O. R. Zaiane, "Tracking changes in dynamic information networks," in *Proc. IEEE Int. Conf. Comput. Aspects Social Netw.*, Oct. 2011, pp. 94–101.
- [129] N. Veldt, D. Gleich, and A. Wirth, "A correlation clustering framework for community detection," in *Proc. World Wide Web Conf. (WWW)*, Lyon, France, Apr. 2018, pp. 439–448.
- [130] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
- [131] Y. Wei and C. Cheng, "Ratio cut partitioning for hierarchical designs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 10, no. 7, pp. 911–921, Jul. 1991.
- [132] D. Watts, "Networks, dynamics, and the small-world phenomenon," *Amer. J. Sociol.*, vol. 105, no. 2, pp. 493–527, 1999.
- [133] P. Wu and L. Pan, "Multi-objective community detection based on memetic algorithm," *PLoS ONE*, vol. 10, no. 5, May 2015, Art. no. e0126845.
- [134] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using neighborhood-inflated seed expansion," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1272–1284, May 2016.
- [135] J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," in *Proc. 16th Pacific-Asia Conf. Knowl. Discovery Data Mining (PAKDD)*, Malaysia, May 2012, pp. 25–36.
- [136] J. J. Xu and H. Chen, "CrimeNet explorer: A framework for criminal network knowledge discovery," *ACM Trans. Inf. Syst.*, vol. 23, no. 2, pp. 201–226, Apr. 2005.
- [137] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, p. 43, 2013.
- [138] K. S. Xu and A. O. Hero, "Dynamic stochastic blockmodels for time-evolving social networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 552–562, Aug. 2014.
- [139] X. Gao, Q. Zheng, D. A. Vega-Oliveros, L. Anghinoni, and L. Zhao, "Temporal network pattern identification by community modelling," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, Dec. 2020.
- [140] J. Xie, M. Chen, and B. K. Szymanski, "LabelRankT: Incremental community detection in dynamic networks via label propagation," in *Proc. Workshop Dyn. Netw. Manage. Mining*, 2013, p. 25.
- [141] K. S. Xu, M. Kliger, and A. O. Hero, *Tracking Communities in Dynamic Social Networks* (Lecture Notes in Computer Science), vol. 6589, J. Salerno, S. J. Yang, and D. Nau, and S.-K. Chai, Eds. Springer, 2011, pp. 219–226.
- [142] B. Yang, D. Liu, J. Liu, and B. Furht, *Discovering Communities From Social Networks: Methodologies and Applications*. New York, NY, USA: Springer, 2010.
- [143] L. Yuan, L. Qin, W. Zhang, L. Chang, and J. Yang, "Index-based densest clique percolation community search in networks," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 5, pp. 922–935, May 2018.
- [144] F. Ye, C. Chen, Z. Zheng, R.-H. Li, and J. X. Yu, "Discrete overlapping community detection with pseudo supervision," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Beijing, China, Nov. 2019, pp. 708–717.
- [145] K. R. Žalik and B. Žalik, "Node attraction-facilitated evolution algorithm for community detection in networks," *Soft Comput.*, vol. 23, no. 15, pp. 6135–6143, Aug. 2019.
- [146] Y. Zhang, Y. Zhang, Q. Chen, Z. Ai, and Z. Gong, "True-link clustering through signaling process and subcommunity merge in overlapping community detection," *Neural Comput. Appl.*, vol. 30, no. 12, pp. 3613–3621, Dec. 2018.



**KAMAL TAHA** (Senior Member, IEEE) received the Ph.D. degree in computer science from The University of Texas at Arlington, USA. He was an Engineering Specialist with Seagate Technology (Computer Disc Drive Manufacturer), USA, from 1996 to 2005. He was also an Instructor of computer science with The University of Texas at Arlington, from August 2008 to August 2010. He has been an Associate Professor with the Department of Electrical and Computer Engineering, Khalifa University, United Arab Emirates, since 2010. He has over 100 refereed publications that have appeared in prestigious top ranked journals, conference proceedings, and book chapters. Over 30 of his publications have appeared with the IEEE TRANSACTIONS journals. His research interests span information retrieval, data mining, databases, defect characterization of semiconductor wafers, bioinformatics, and information forensics and security, with an emphasis on making data retrieval and exploration in emerging applications more effective, efficient, and robust. He serves as a member for the program committee, the editorial board, and the Review panel for several international conferences and journals.

...